On the Designs and Challenges of Practical Binary Dirty Paper Coding

04 / 08 / 2009

Gyu Bum Kyung and Chih-Chun Wang
Center for Wireless Systems and Applications
School of Electrical and Computer Eng.

PURDUE UNIVERSITY

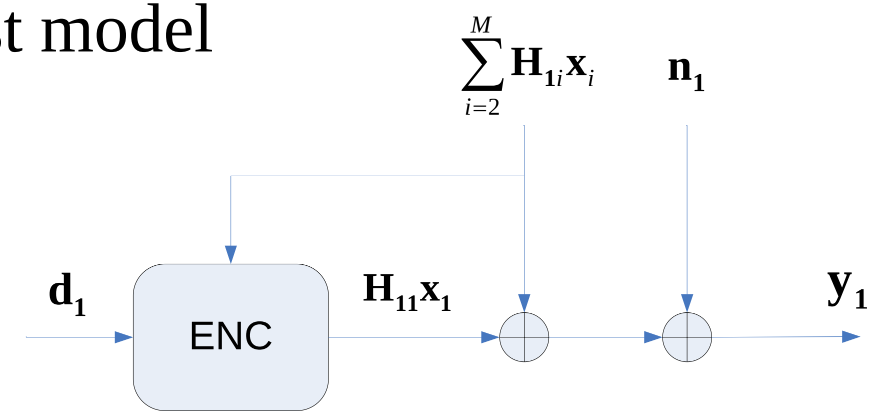
Outline

- Introduction
- General Framework of Binary DPC
- Binary DPC-The Proposed Practical System
  - The system model
  - Decoding in the Factor Graph
- Code Design
  - Choose the system parameters
  - Code optimization
- Simulation and Discussions
- Conclusion

Introduction

- Multi-user MIMO broadcast model

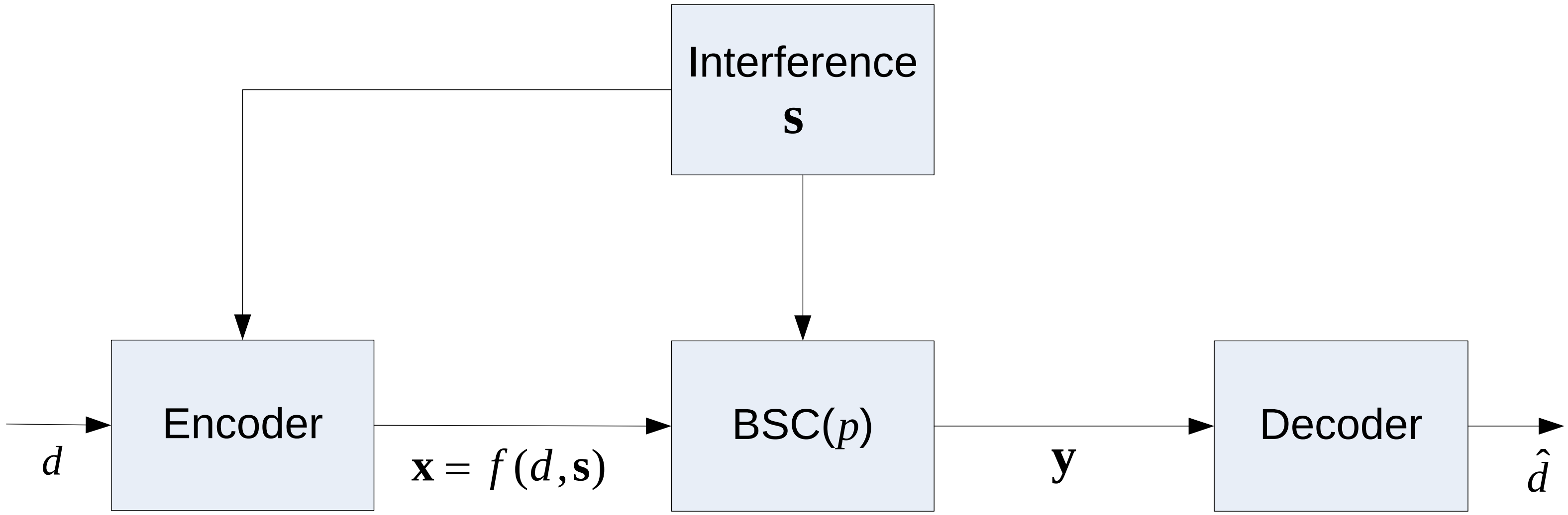
$$\mathbf{y}_1 = \mathbf{H}_{11}\mathbf{x}_1 + \sum_{i=2}^{M}\mathbf{H}_{1i}\mathbf{x}_i + \mathbf{n}_1$$



- Why dirty paper coding (DPC)?
  - Eliminating inter-user interference (IUI)
  - Alternative approach: linear processing (away from the capacity)
  - Costa's proof (Costa '83)
  - Footstone of theoretical studies of the Gaussian MIMO broadcast channel (Caire *et al.* '03)

General Framework of Binary DPC

- DPC encoder sends the transmitted signal x, a function $f(d,s)$ of the interference s and information data d
- $y = x + s + n$
- The goal of DPC is to optimize the transmission rate R subject to a normalized power constraint W on x.

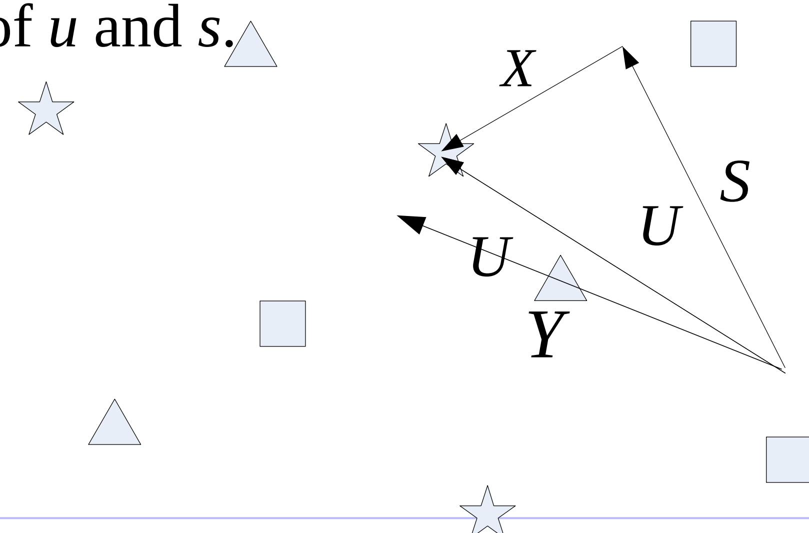


Achieving DPC capacity: Random binning

- Noncausal side information (Gel'fand and Pinsker '80)

$$C = \max_{p(u,x|s)} \left[I(U;Y) - I(U;S) \right].$$

- Generate $2^{N(I(U;Y)-\delta)}$ sequences, divide these into 2^{NR} bins
- R is chosen to be less than $I(U;Y) - I(U;S) - \varepsilon$, each bin has $2^{N(I(U;S)-\varepsilon')}$ sequences
- Encoder finds a sequence *u* which is the closest to *s* based on *d* and sends *x* which is a function of *u* and *s*.
- Decoder looks for a sequence *u* s.t. *u* and *y* are the closest and identifies the bin which includes *u*.

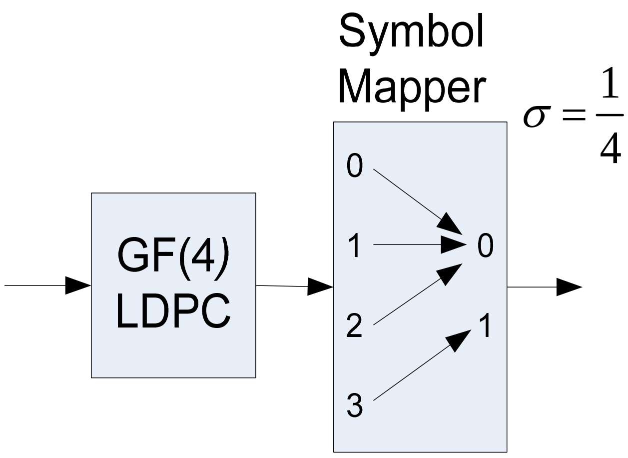


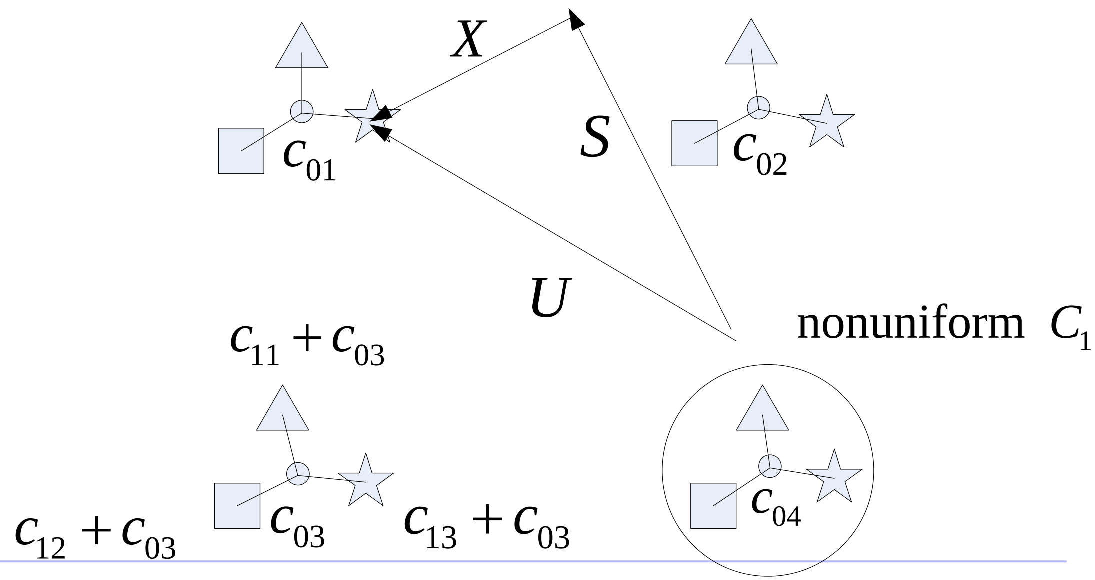
Coset-based binning

- C_0, C_1 : the quantization code and the information bearing code
- d is mapped to c_1 , $c_1 + C_0$ forms a coset (a bin)
- Given s, the encoder finds c_0 such that $c_1 + c_0$ in the $c_1 + C_0$ bin is the closest to s
- The transmitted signal x is $c_0 + c_1 - s$
- The decoder finds c_0 and c_1 such that $c_0 + c_1$ is the closest to y.
- For a practical implementation, if c_0 and c_1 are uniform, then no bit-based message-passing decoder can extract any information from $c_0 + c_1$

Superpostion-coding-based (Bennatan *et al.* '06)

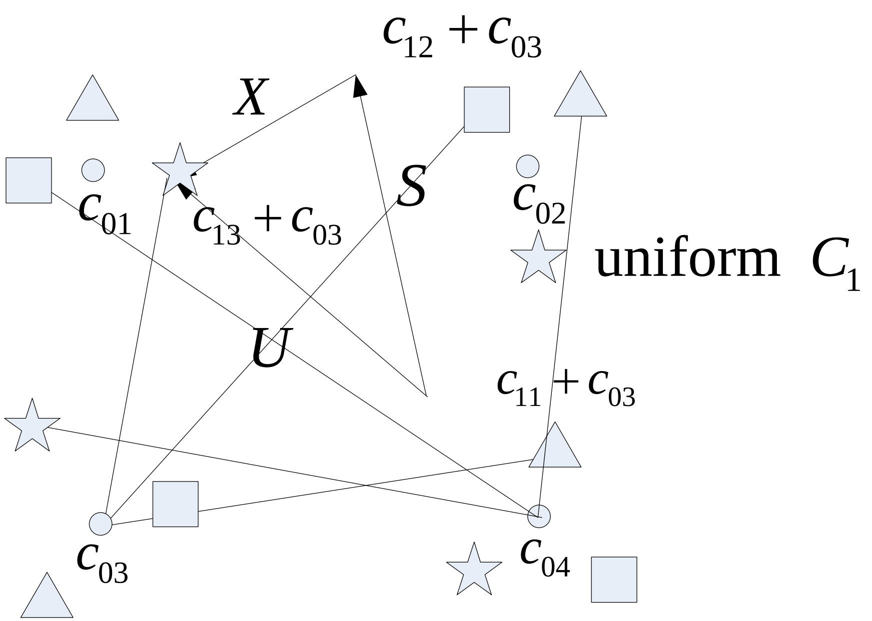
- Either c_0 or c_1 has non-uniform a priori distribution to initialize iterative decoding
- Nonbinary low-density parity-check (LDPC) codes
  - The normalized Hamming weight σ of c_1
  - GF(q) LDPC code can only achieve $\sigma = {}^{1}/_{q}, {}^{2}/_{q}, \ldots, {}^{(q-1)}/_{q}$
- Symbol mapper – a nonlinear code





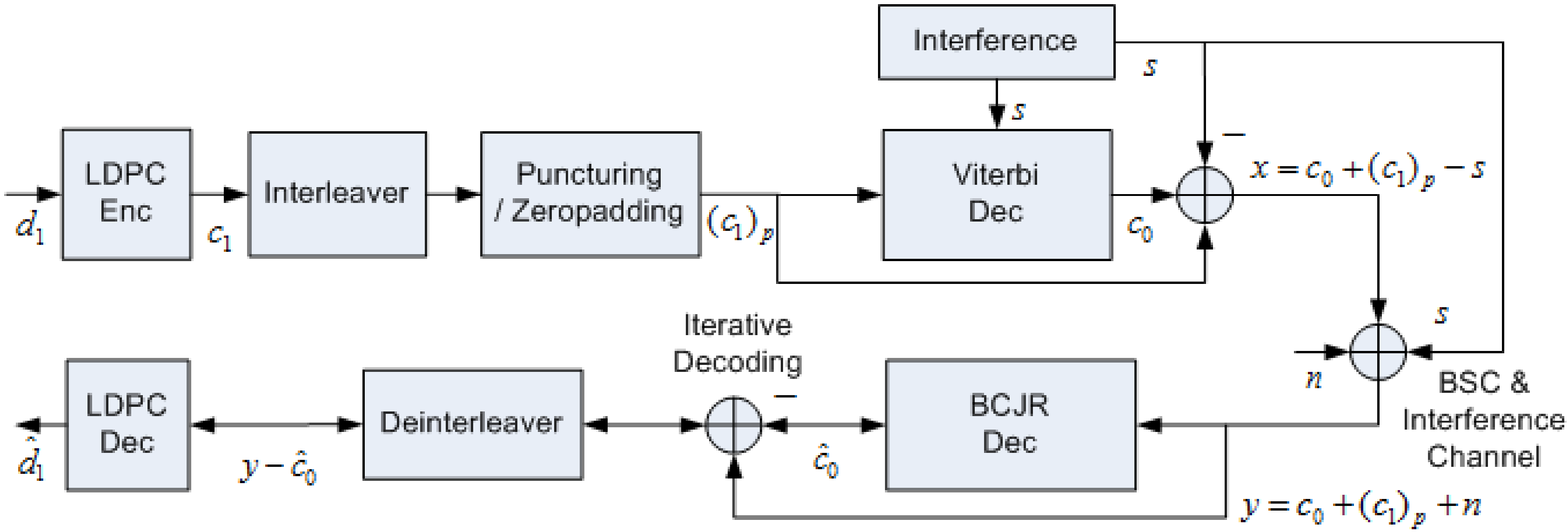
The proposed system

- Coset-based binning
  - Initialization problem
- ***Edge erasing*** with ***binary*** LDPC codes to initialize iterative decoding
- Code optimization
  - Density evolution (DE), the extrinsic information transfer (EXIT) chart



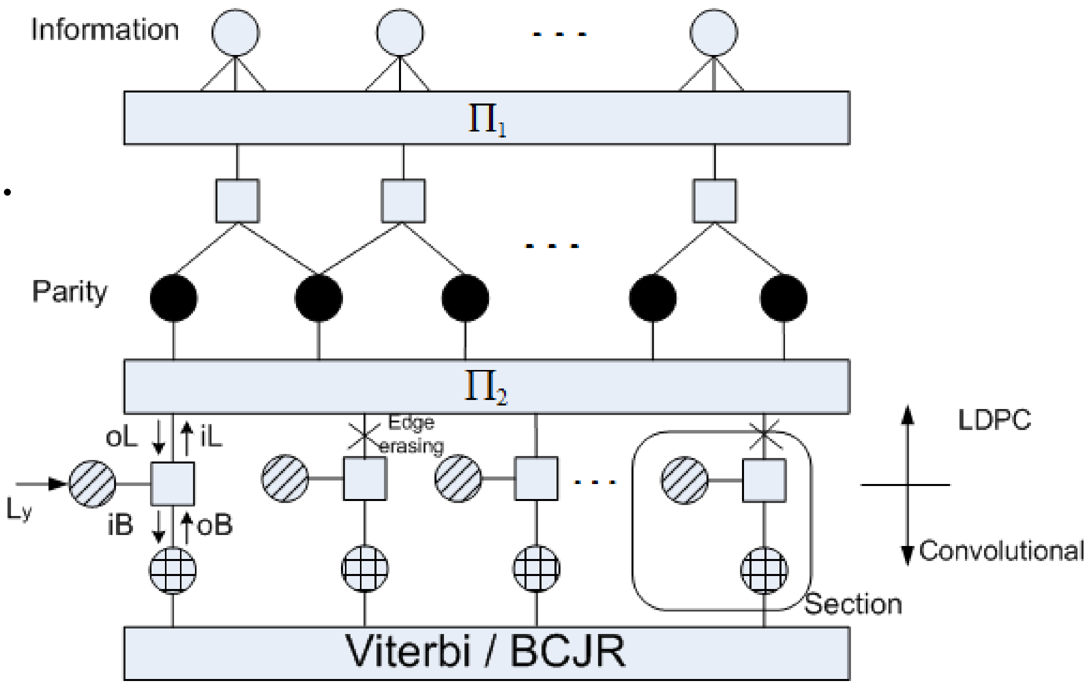
System model

- Random interleaver to reduce the dependence
- c_1 is punctured by *e* and is padded with zeros.
- Viterbi decoder chooses c_0 such that $c_0 + (c_1)_p$ is closest to s
- $y = x + s + n = c_0 + (c_1)_p + n$



Decoding in the Factor Graph

- Iterative decoding between the BCJR and LDPC decoders
- Set LLR values at punctured positions using $y = c_0 + n$
  - BCJR decoder - Log MAP
- After BCJR decoding, the extrinsic information in the non-punctured part is delivered to LDPC decoder.
- LDPC decoding is performed with the received bits and the extrinsic LLR information
- Continue iterative decoding



System parameters

- BSC with p and weight constraint W are given and design parameters are R_0 and R_1

	The proposed scheme	Superposition (Bennatan *et al.*)
R_0 and R_1	$R_0 > 1 - h(W),\ R_1 < h(W) - h(p)$	
W and p	No constraint	$W = \sigma(1-p) + p(1-\sigma)$
q	No constraint	$\sigma = 1/q$ to $(q\text{-}1)/q$

- To flexibly support different W values, very high-order GF(q) has to be used - increases complexity
- Our system is flexible and can easily handle different weight constraints W using *edge erasing* and *binary* LDPC codes.

Code optimization

- The EXIT chart
  - Use *oL* and *iL* as the input and the output
  - Assuming *oL* (*iL* resp.) is always Gaussian distributed input with mean and variance (μ , 2μ) for BCJR (LDPC)
  - The mutual information of the output LLR messages is obtained by

$$I(X;Y) = h(X) - h(X\,|\,Y) = 1 - h(X\,|\,Y)$$

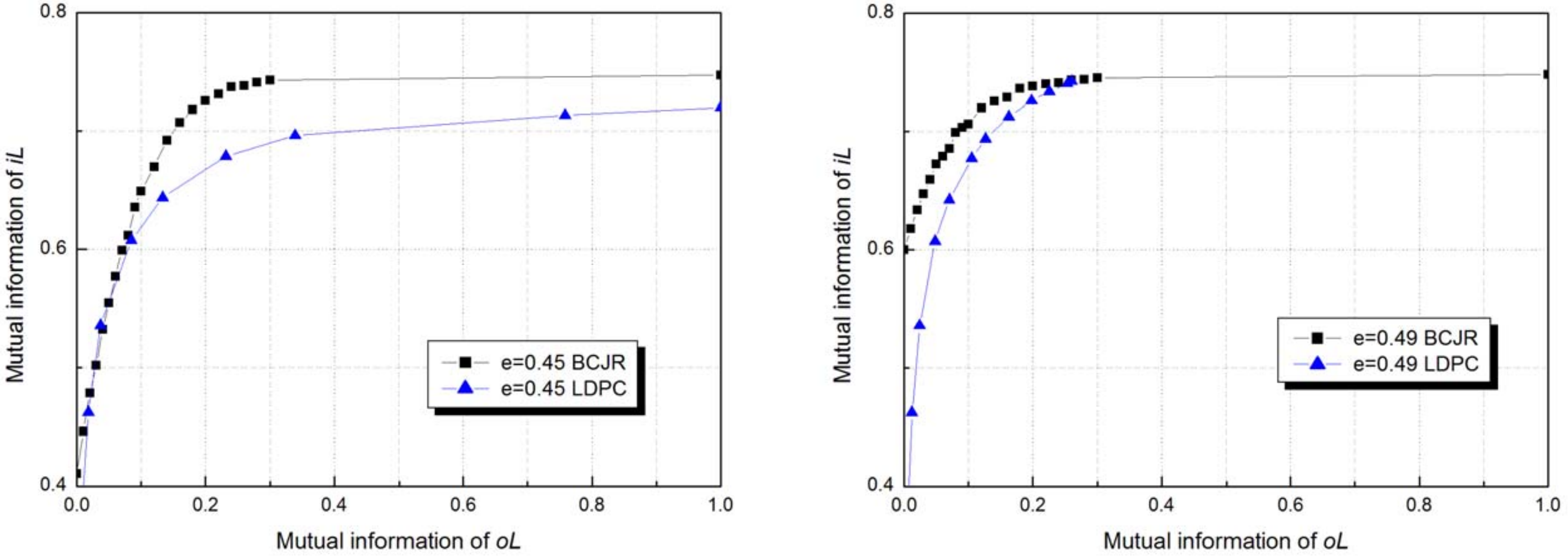
$$= 1 - \int_{-\infty}^{+\infty} \log_2 \frac{e^m + 1}{e^m} P(m\,|\,X = 0)\,dm$$

where m denotes the LLR messages, that is, $m = \frac{\log_2(Y\,|\,X = 0)}{\log_2(Y\,|\,X = 1)}$

PURDUE UNIVERSITY

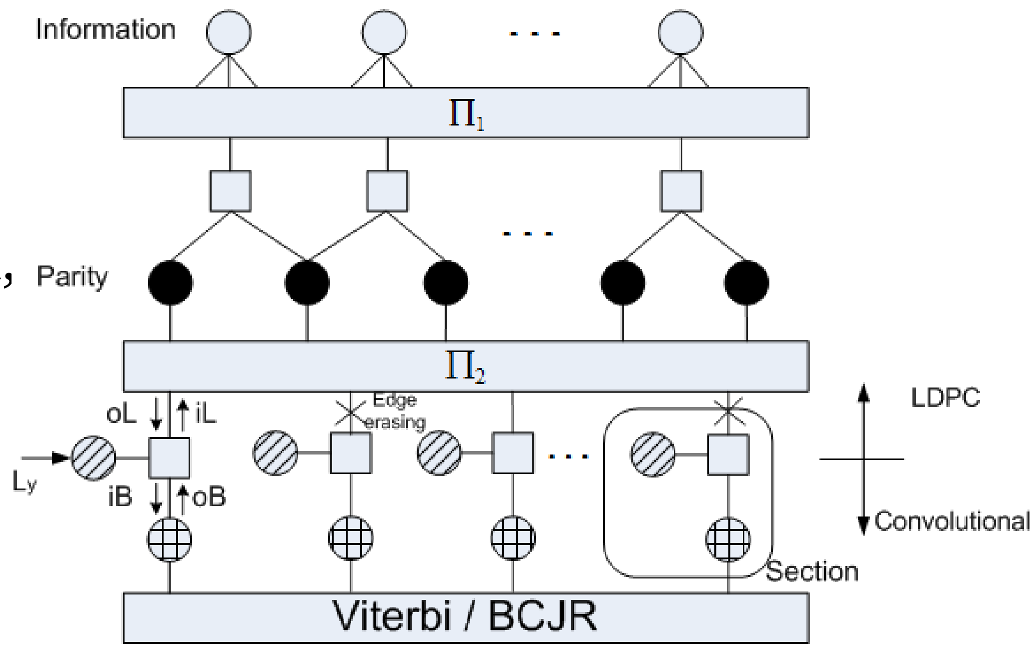
How to choose e

- Estimate the threshold p^*
  - By selecting the largest p value - two curves do not cross each other
- Optimize the e value
  - Given p^*, choose an e value - the two curves are the farthest apart
- Optimize p^* and e iteratively



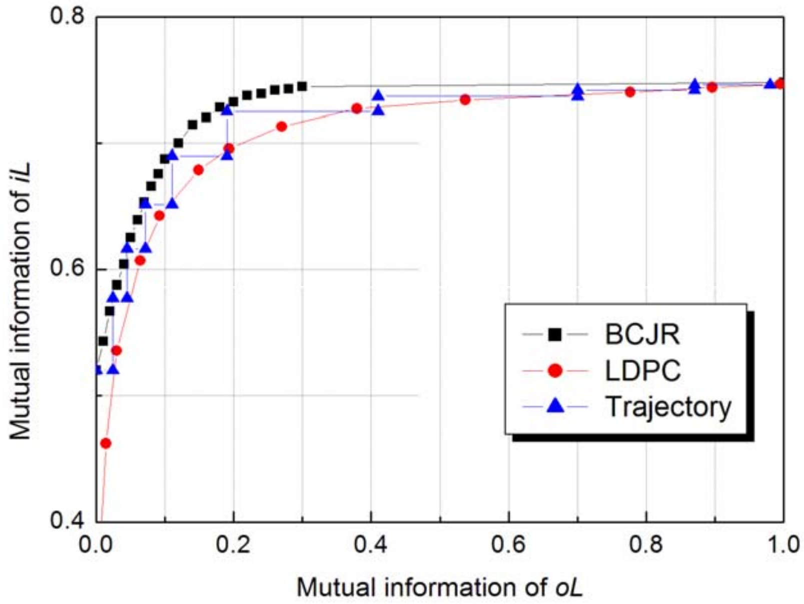
Code design for LDPC code

- The joint use of DE and the EXIT chart
- Record the distributions of *iL*, use the pmfs as an input of DE
- Perform DE iterations, obtain the distribution of *oL*
- By computing the mutual information of *oL*, use the EXIT curve of BCJR to find the pmf of *iL*
- Given a degree distribution, check whether the EXIT curve and DE converge
- Differential Evolution



Trajectory of Monte Carlo Simulation

- Record the LLR distributions under real BCJR+LDPC decoding and convert them to the mutual information
- The zigzag trajectory fits the two EXIT curves well
- By choosing different numbers of LDPC iterations in the initial and final stages, we can control the total number of LDPC iterations to be roughly 100-150
- How many outer loop iteration?



Simulation and Discussions

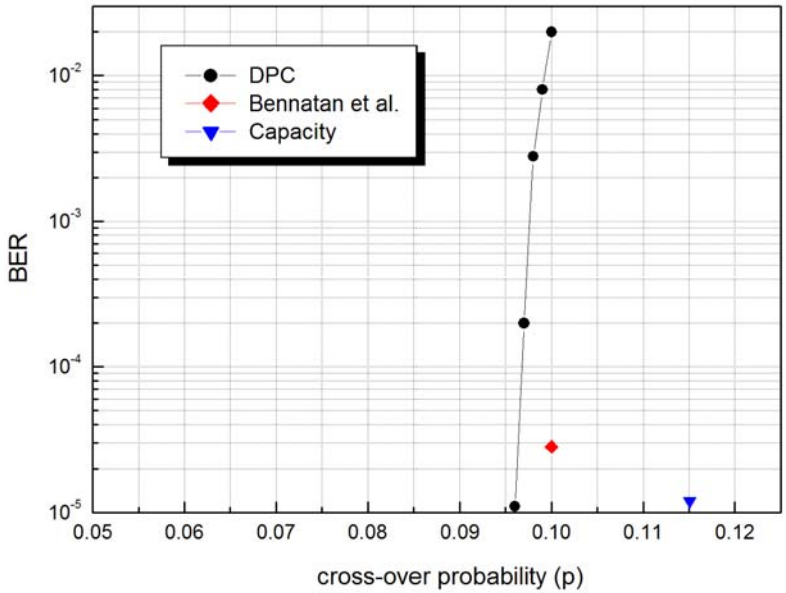
- Quasi-cyclic LDPC codes ($R_1 = 0.36$, $N = 10^5$)
  - 200×272 base matrix
  - $\lambda(x) = 0.53x + 0.21x^2 + 0.01x^3 + 0.25x^9, \ \rho(x) = 0.2x^2 + 0.8x^3$
- Convolutional code ($R_0 = 1/8$)

  $(2565, 2747, 3311, 3723, 2373, 2675, 3271, 2473)$

- Performance

  $p^* = 0.097$ vs $p^* = 0.1$

- Our scheme have several advantages – complexity, flexibility, and efficient encoding and decoding.



Conclusion

- Practical scheme for binary dirty-paper channels based on random binning
- Binary LDPC codes and edge erasing - the advantages for complexity.
- Choose system parameters flexibly - important in the practical system.
- Code design combining EXIT chart and DE jointly
- Similar performance to that of state-of-the-art superposition-coding-based binary DPC scheme.
- Future research – Extend to Gaussian DPC