Design of Coarsely-Quantized Message Passing Decoders

Gerhard Kramer (TUM)

Joint work with Gottfried Lechner (UniSA) and Troels Pedersen (Aalborg)
Contributions by Emna Ben Yacoub (TUM)

Oberpfaffenhofen Workshop on High Throughput Coding
February 27, 2019

Binary Message Passing (BMP) for LDPC Codes

- Gallager, “Low density parity check codes,” IRE Trans. IT, 1962
- Kou, Lin, Fossorier, LDPC codes based on finite geometries, IEEE Trans. IT, 2001
- Zhang, Fossorier, Modified bit-flipping decoding, IEEE Comm. Lett., 2004
- Miladinovic, Fossorier, Improved bit-flipping decoding, IEEE Trans. IT, 2005
- Jiang, Zhao, Shi, Chen, Improved bit flipping decoding, IEEE Comm. Lett., 2005
- Ardakani, Kschischang, Properties of binary message-passing, IEEE Trans. IT, 2005
- Sankaranarayanan et al., Failures of the Gallager B decoder, ISIT 2006
- Reliability-based majority-logic decoding for LDPC codes, IEEE Trans. Comm., 2009
- Planjery, Declercq, Danjean, Vasic, Finite alphabet iterative decoders, 2013-
- Many other papers

Here Review and Expand on:

- Lechner, Pedersen, Kramer, Analysis and design of binary message passing decoders, IEEE Trans. Comm., 2012 (and ISIT 2007)

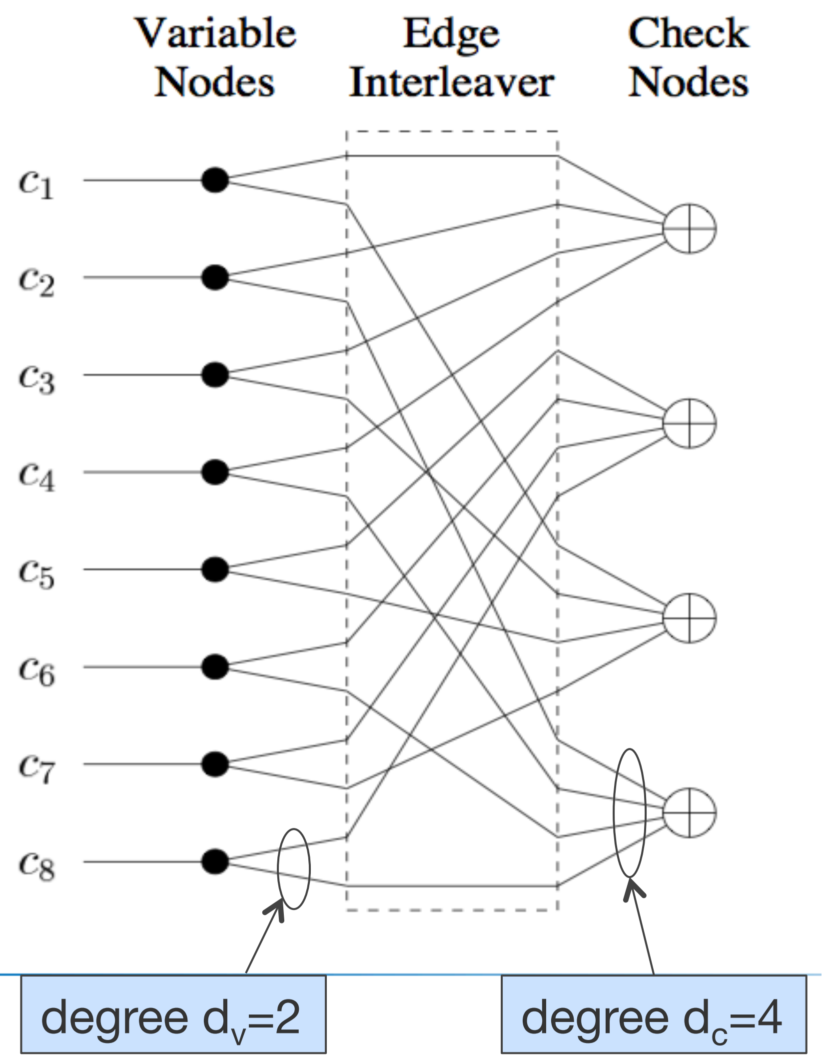
I. Low-Density Parity-Check (LDPC) Codes

- A binary linear block code is the set of binary (row) vectors, or codewords, $\underline{c}$, satisfying, e.g.,

$$\underline{c} \underbrace{\begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}}_{\text{matrix } H^T} = \underline{0}$$

where H is a (n-k) x n parity-check-matrix.
Rate is R=1-rank(H)/n (example: R=5/8).

Tanner Graph Representation of Parity-Check Constraints



- Code is low-density if each row and column of H^T has “few” 1’s
- Irregular LDPC code: variable number of 1’s in every column/row
- Decoding: use message passing on the graph
- Messages may be cond. probabilities

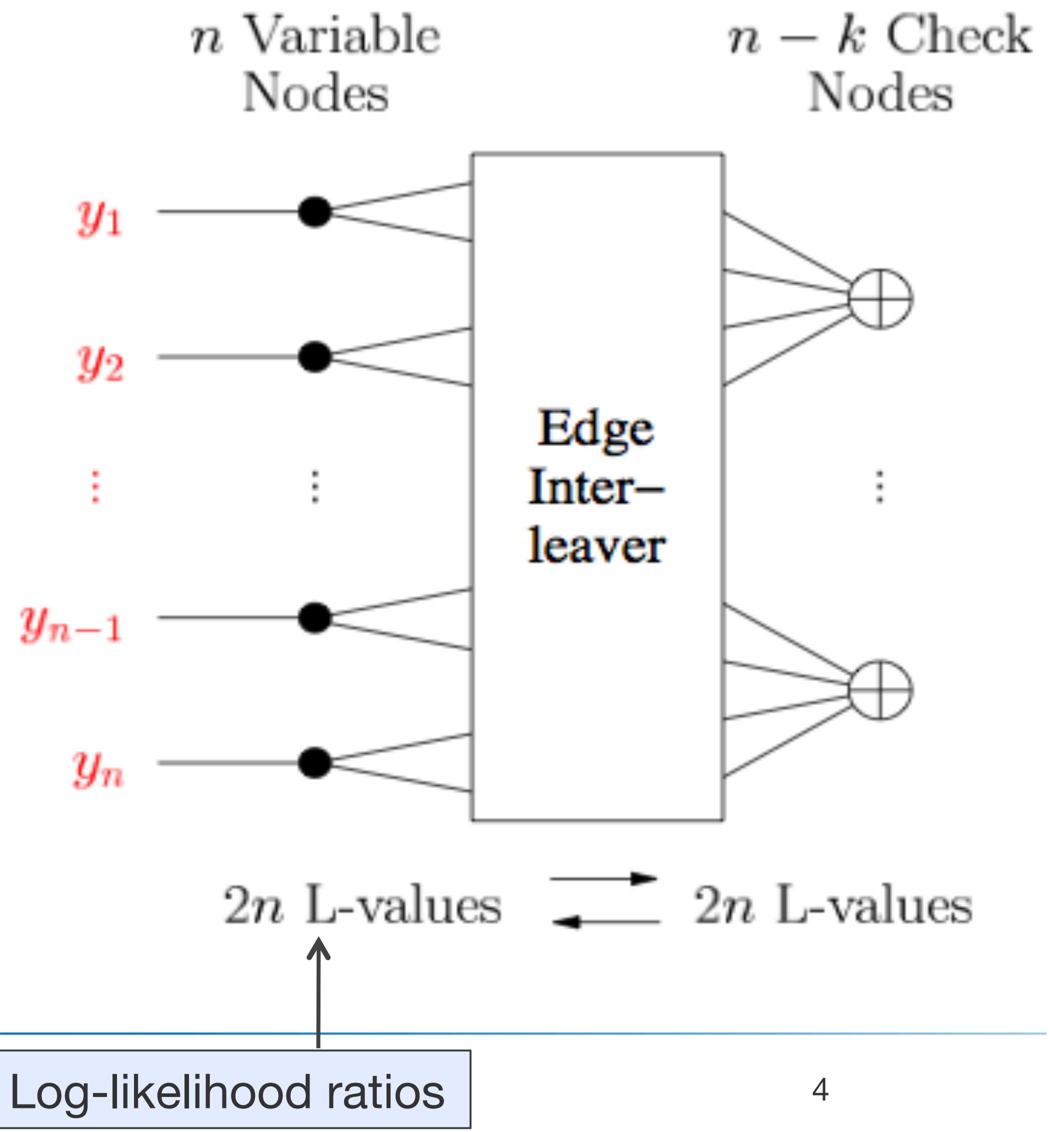
$$\Pr\left(c_1 = 0 \middle| \underline{y}\right)$$

or log-likelihood ratios (L-values)

$$L_1 = \log \frac{\Pr\left(c_1 = 0 \middle| \underline{y}\right)}{\Pr\left(c_1 = 1 \middle| \underline{y}\right)}$$

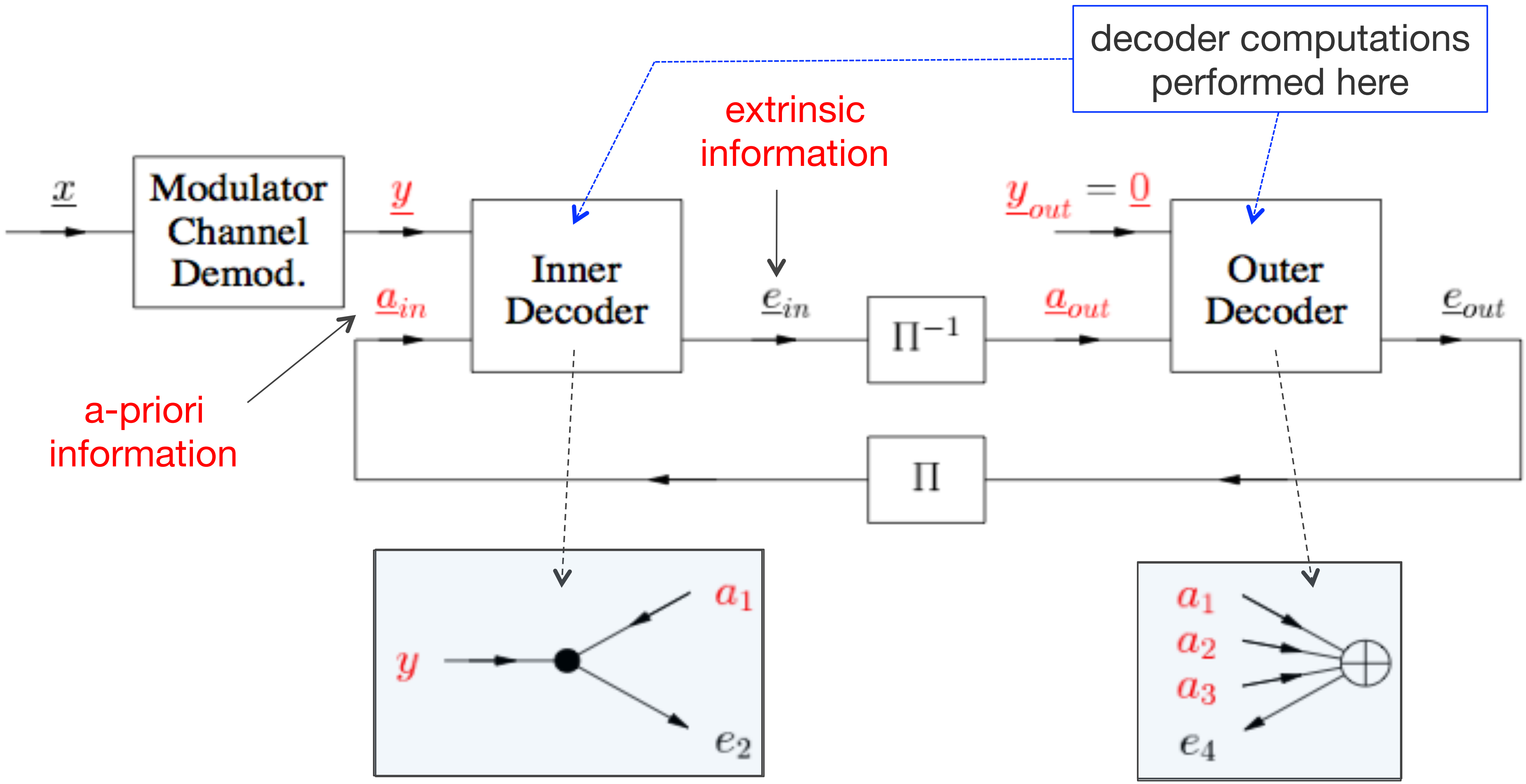
or, in practice, quantized L-values

Message Passing



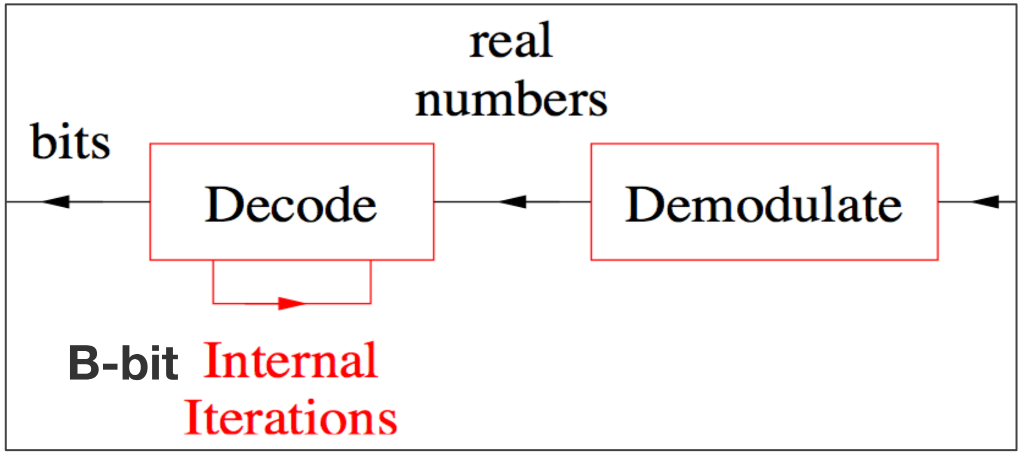
II. Iterative Decoding

LDPC code decoder iterations (turbo processing):

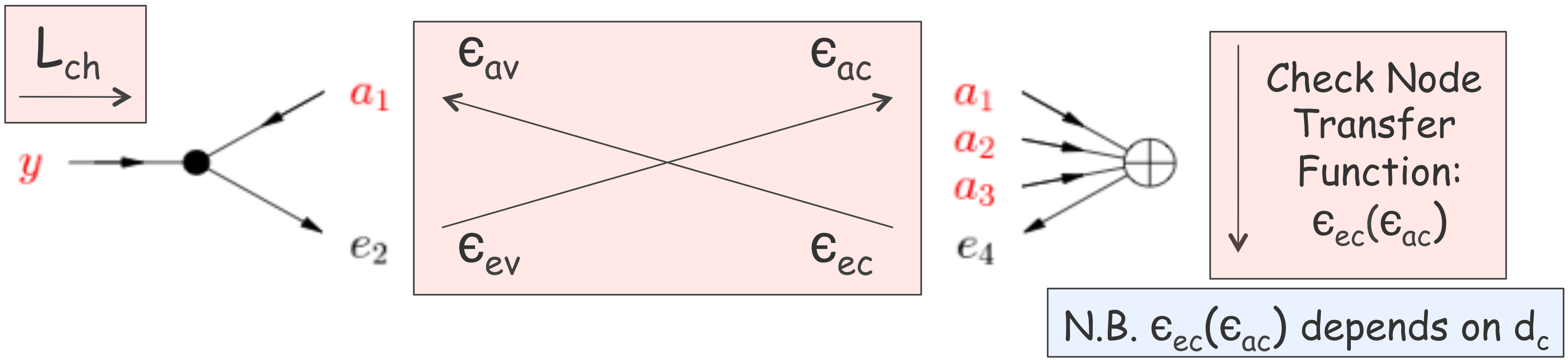


III. Demodulation and Decoding

- L-values are real but must be quantized, see figure below
    1) Demodulator: can put out soft decisions (>$\log_2(M)$ bits/symbol) or hard decisions (=$\log_2(M)$ bits/symbol)
    2) Decoder iterations: B-bit message passing:
       binary message passing (BMP, B=1)
       ternary message passing (TMP, B≈2)
- Motivation: high-speed devices (>100 Gb/s) need simplifications



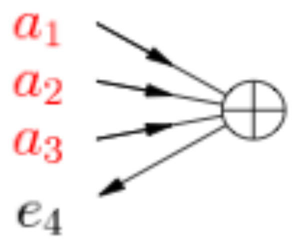
- BMP/TMP: natural approaches are as follows:
  1) Every edge bit represents a hard decision on an extrinsic L-value
  2) Variable nodes convert apriori bits to L-values, add L-values, and make binary (hard) or ternary decisions on output L-values
  3) Check nodes perform (extrinsic) XORs for binary message passing, and (extrinsic) XORs and erasures for ternary message passing
- Analysis: use distribution evolution (DE) to track extrinsic probabilities. BMP: track error probabilities; TMP: also track erasures



BMP Distribution Evolution

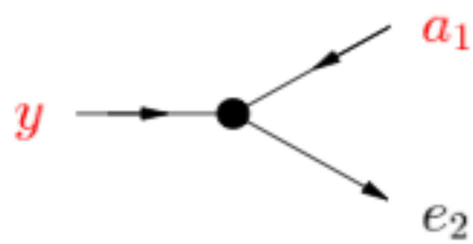
- Check node (degree d_c) and binary messages:

$$\epsilon_{ec} = f_c(\epsilon_{ac}; d_c) = \frac{1-(1-2\epsilon_{ac})^{d_c-1}}{2}$$



- Variable node (degree d_v): suppose $x_j=\pm1$ (BPSK)

$$\varepsilon_{ev} = \sum_{j=1}^{d_v} \frac{1}{d_v} \Pr\left[\operatorname{sgn}\left(L_{ev,j}\right) \neq x_j\right]$$



$$L_{ev,j} = L_{ch} + \sum_{i=1;i\neq j}^{d_v} L_{av,i} \qquad j = 1, 2, \ldots, d_v$$

$$L_{av,i} = a_i \log\frac{1-\varepsilon_{av}}{\varepsilon_{av}}, \ a_i = \pm 1, \text{ but what is } \varepsilon_{av}\ ?$$

L_{ch} depends on channel quantization

Two design issues

Issue 1: Variable Node Processing

- Processing depends on ϵ_{av} which
  - Varies from iteration to iteration
  - Is unknown, unless the codes have infinite length in which case ϵ_{av} can be computed from EXIT chart (see below)
- Two other approaches:
  - Optimize "choice" of ϵ_{av} offline by numerical simulation
  - Estimate ϵ_{av} online based on the number of unsatisfied checks
- 1^{st} approach is complex, but likely very good. This variant was used to design certain deployed LDPC codes
- 2^{nd} approach is used here

Issue 2: Channel Outputs

- Consider an AWGN channel, $x_j=\pm 1$, noise variance σ_n^2
- Let $D_{ch}=|L_{ch}|$ … called the reliability of the L-value
- For soft decisions:

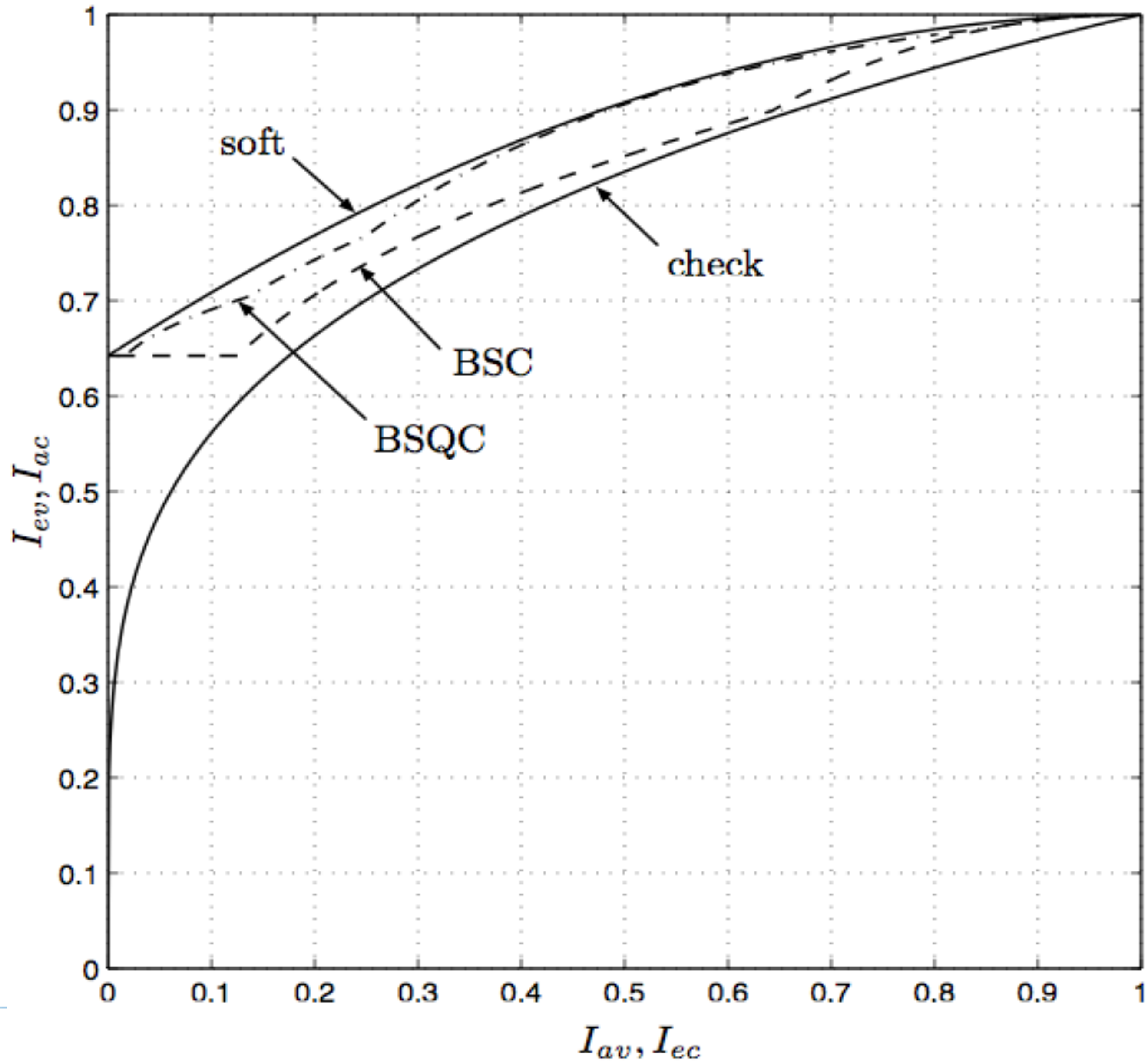
$$D_{ch} = \frac{2}{\sigma_n^2}$$

- For hard decisions get a binary symmetric channel (BSC) with crossover probability ϵ_{ch} $(0 \le \epsilon_{ch} \le 0.5)$

$$D_{ch} = \log \frac{1-\varepsilon_{ch}}{\varepsilon_{ch}}, \text{ where } \varepsilon_{ch} = Q\left(1/\sigma_n\right)$$

- For b-bit quantization: use mixture of b hard-decision channel reliabilities, e.g., 2-bit quantization with a binary symmetric quaternary output (BSQC) channnel

Transfer Functions for (4,6)-Regular Code



Channel: σ_n=0.67
x-axis to y-axis:
variable nodes
y-axis to x-axis:
check nodes

$I_{ac}=1-h(\epsilon_{ac})$ where h(x) is the binary entropy function:
$h(x) = -x\log_2 x-(1-x)\log_2(1-x)$
Example: h(0.11)=0.5
Similar for I_{ec}, I_{av}, I_{ev}

Comments:

- BSC quantization same as Gallager B algorithm
- BSQC quantization thresholds at 0 and ±1.9

IV. Optimization: Irregular LDPC Codes

- Each node's ϵ_{ev} depends on d_v: write as $\epsilon_{ev}(\epsilon_{av},d_v)$. Now use different degrees to shape avg. variable node curve:

$$\varepsilon_{ev}\left(\varepsilon_{av}\right)=\sum_{i}\lambda_i\,\varepsilon_{ev}\left(\varepsilon_{av},i\right)$$

  with λ_i=fraction of edges connected to var. nodes of degree i
- Can similarly shape the check node function $\epsilon_{ec}(\epsilon_{ac})$
- Degree distribution $\{\lambda_i\}$ design: use EXIT chart
  - $\epsilon_{ev}(\epsilon_{av})$ curve should lie above $\epsilon_{ec}(\epsilon_{ac})$ curve for convergence (and $n=\infty$)
  - L-value messages: Matching EXIT curves maximizes rate.
- BMP: new issues vs. L-value messages
  - Stability (decoder convergence when ϵ_{av} or ϵ_{ac} are small)
  - Cycles related to "absorbing sets" cause decoder to get stuck
- Approach: build optimization & remedies into a linear program

Rate, Stability, Cycles

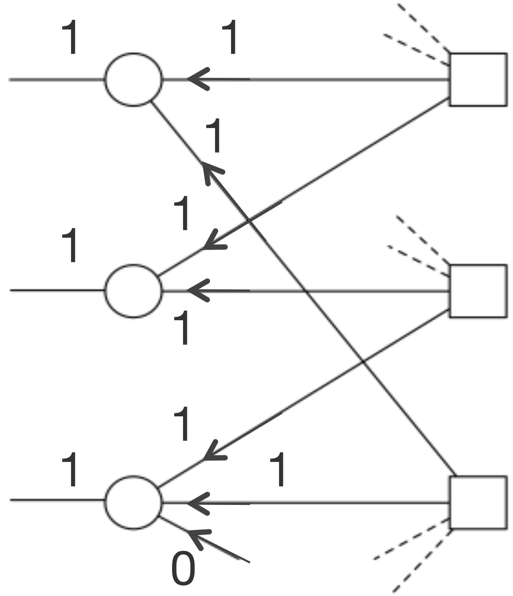
- Design Rate:

$$R = 1 - \frac{1/d_c}{\sum_i \lambda_i / i}$$

- Stability: satisfied for binary message passing and hard or soft channel messages if and only if (try $\lambda_2 = 1$)

$$\left(\lambda_2 + 2\varepsilon_{ch}\lambda_3\right)\left(d_c - 1\right) < 1$$

- Cycles:
  - Structure on right causes decoding failure if all channel messages in error, and if all other incoming messages correct
  - Obvious idea: avoid cycles of degree 2 or 3 variable nodes



Cycles and Linear Program

- Result: a Tanner graph with no cycles having degree 2 and 3 variable nodes exists if and only if (try $\lambda_3 = 1$)

$$3\lambda_2 + 4\lambda_3 \le \frac{6}{d_c}\left(1 - \frac{1}{(1-R)N}\right) < \frac{6}{d_c}$$

- Linear Program: $\lambda = \{\lambda_i\}$ is variable node degree distribution

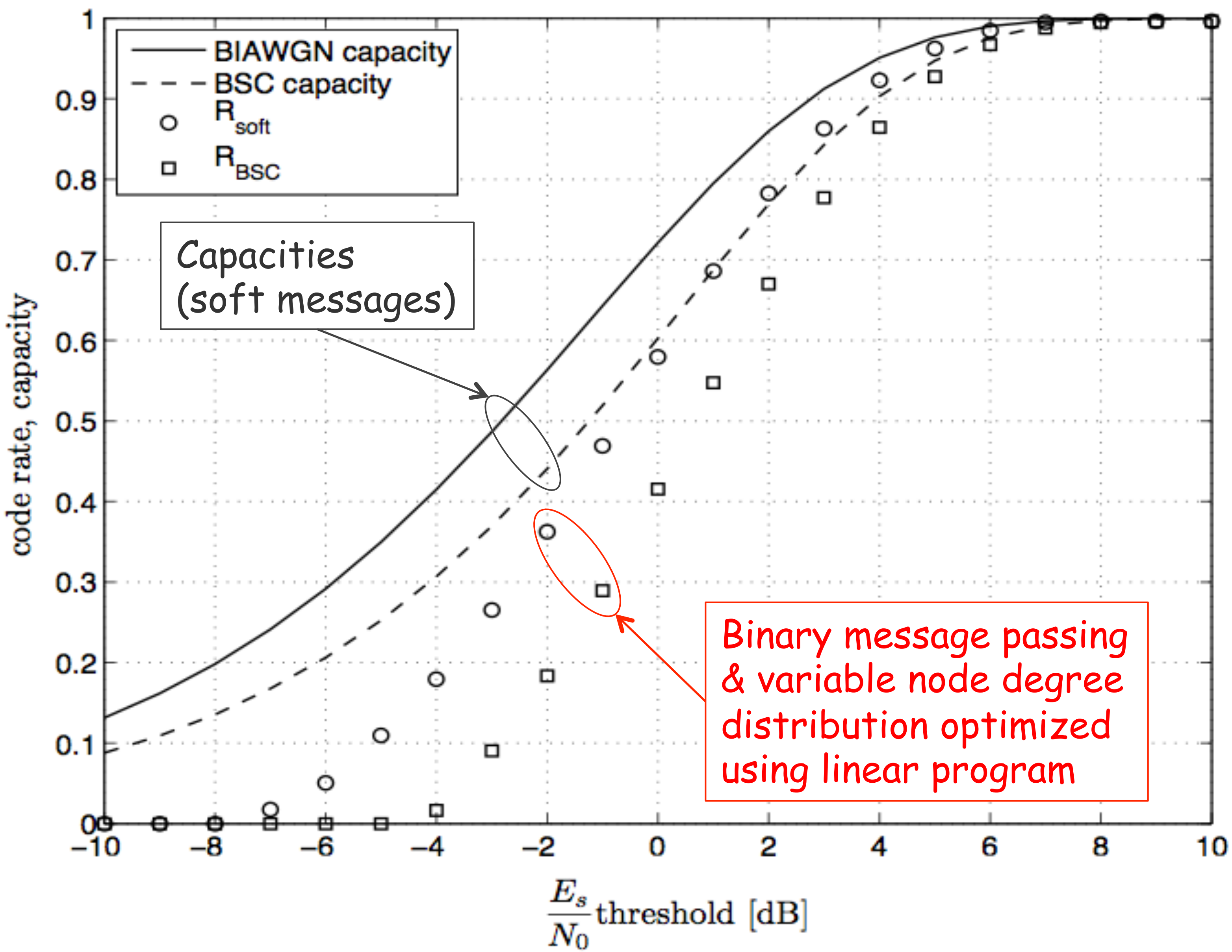
$$\lambda^* = \arg\max_{\lambda} R = \arg\max_{\lambda} \left(1 - \frac{1/d_c}{\sum_i \lambda_i / i}\right) = \arg\max_{\lambda} \sum_i \lambda_i / i$$

subject to [variable node EXIT curve above check node EXIT curve]

$$\sum_i \lambda_i = 1, \quad 0 \le \lambda_i \le 1$$

$$(\lambda_2 + 2\varepsilon_{ch}\lambda_3)(d_c - 1) < 1, \quad 3\lambda_2 + 4\lambda_3 < \frac{6}{d_c}$$

BMP Thresholds

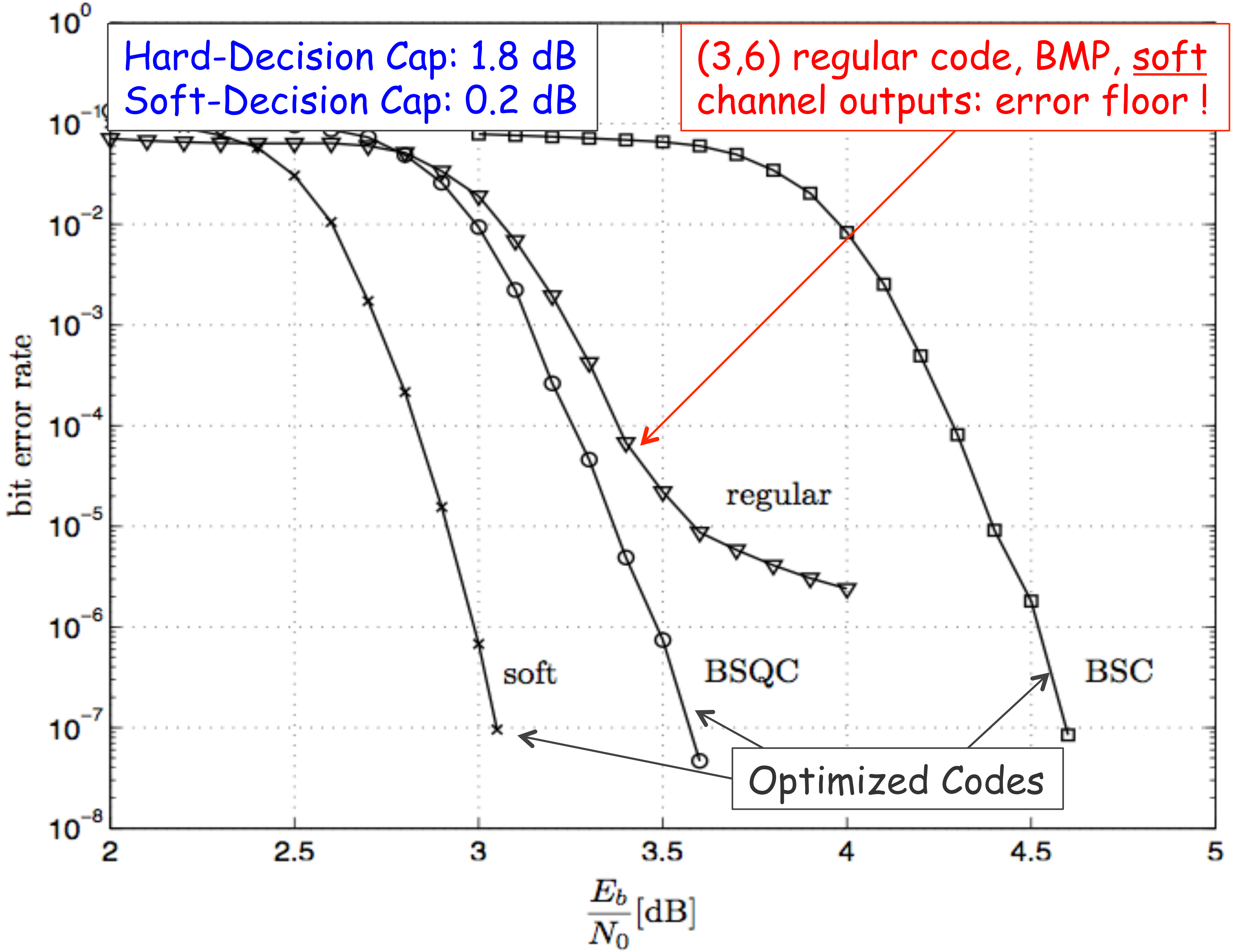


Comments:

- x-axis is E_s/N_0
- hard decision (BSC) capacity ≈2dB below AWGN capacity at low rate
- gap to capacity decreases as rate increases, for hard decisions and BMP

- Conclusion: high rate is good for BMP

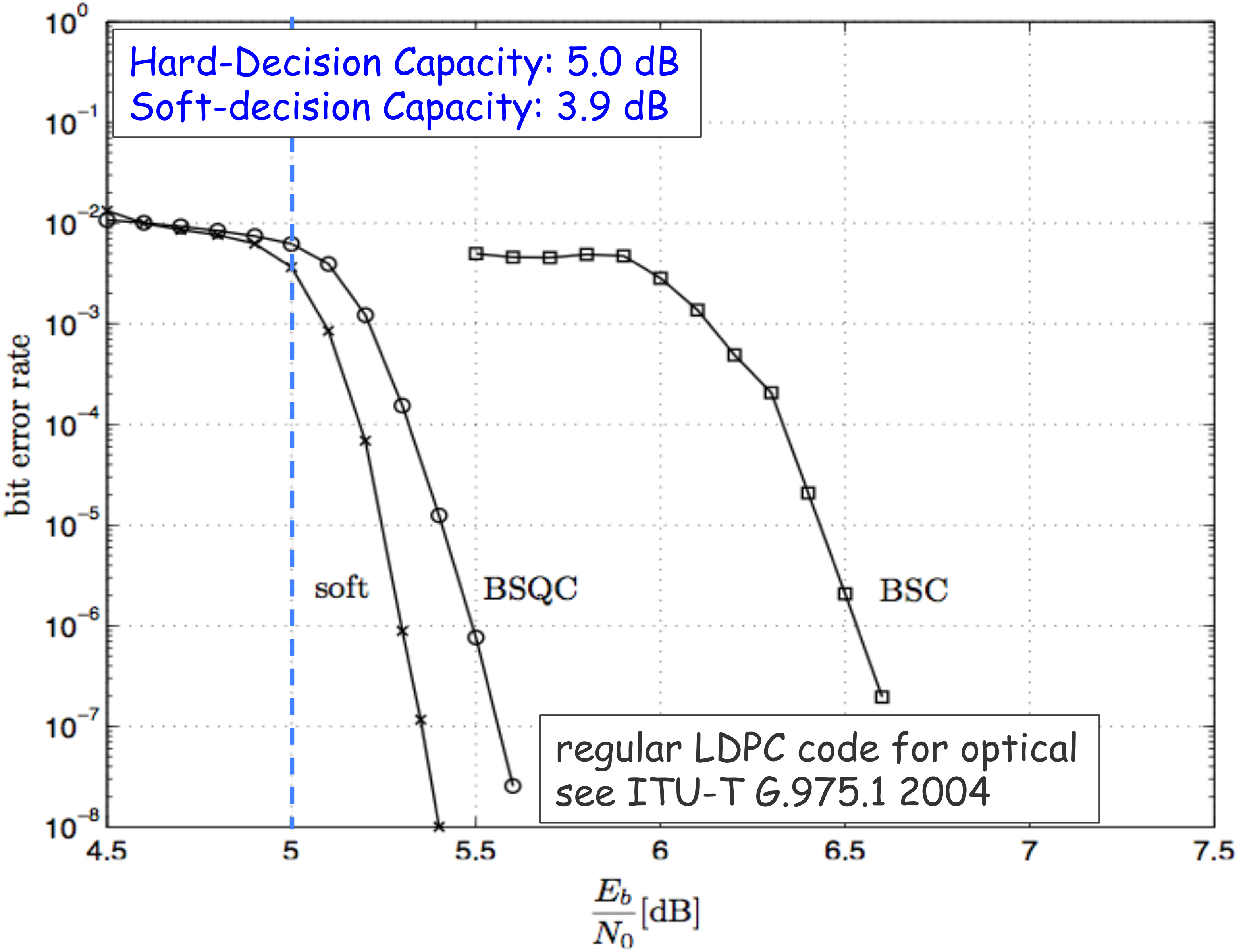
Performance: Rate 1/2, BMP



Comments:

- x-axis is E_b/N_0
- PEG interleavers automatically avoid undesirable cycles
- n = 10,000
- 2-bit quant. gains ≈1dB over Gallager B and loses ≈0.5dB as compared to soft outputs

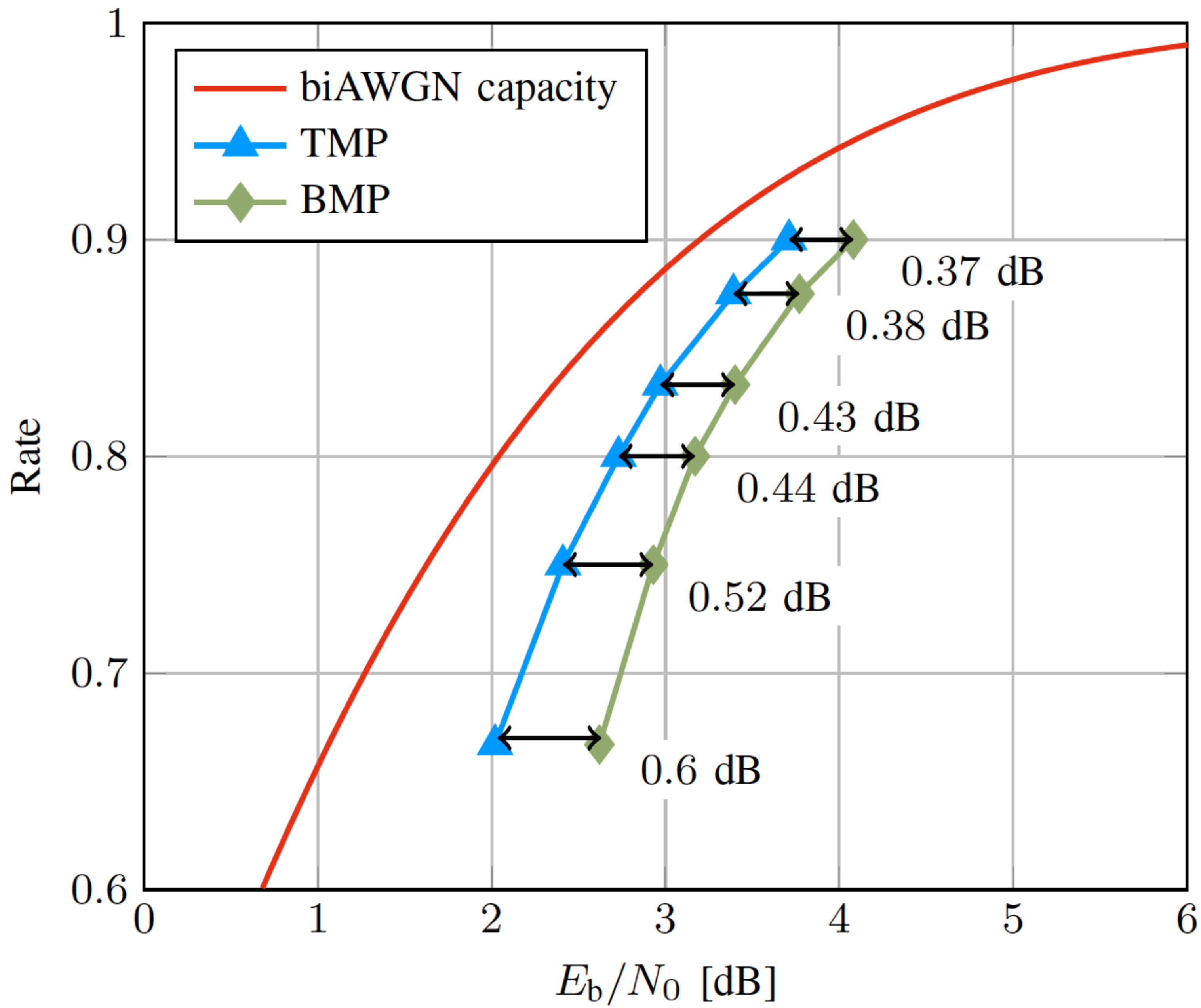
Performance: Rate 15/16, BMP



Comments:

- x-axis is E_b/N_0
- interleaver taken from standard
- 2-bit quant. gains ≈1dB over Gallager B and loses ≈0.2dB vs. soft outputs
- BMP is ≈1.5dB from L-value message capacity
- longer & irregular codes get closer

Performance: TMP



Comments:

- Figure taken from Emna Ben Yacoub's Master Thesis, Oct. 2018
- Curves show decoding thresholds with BMP and TMP for optimized protograph LDPC code ensembles

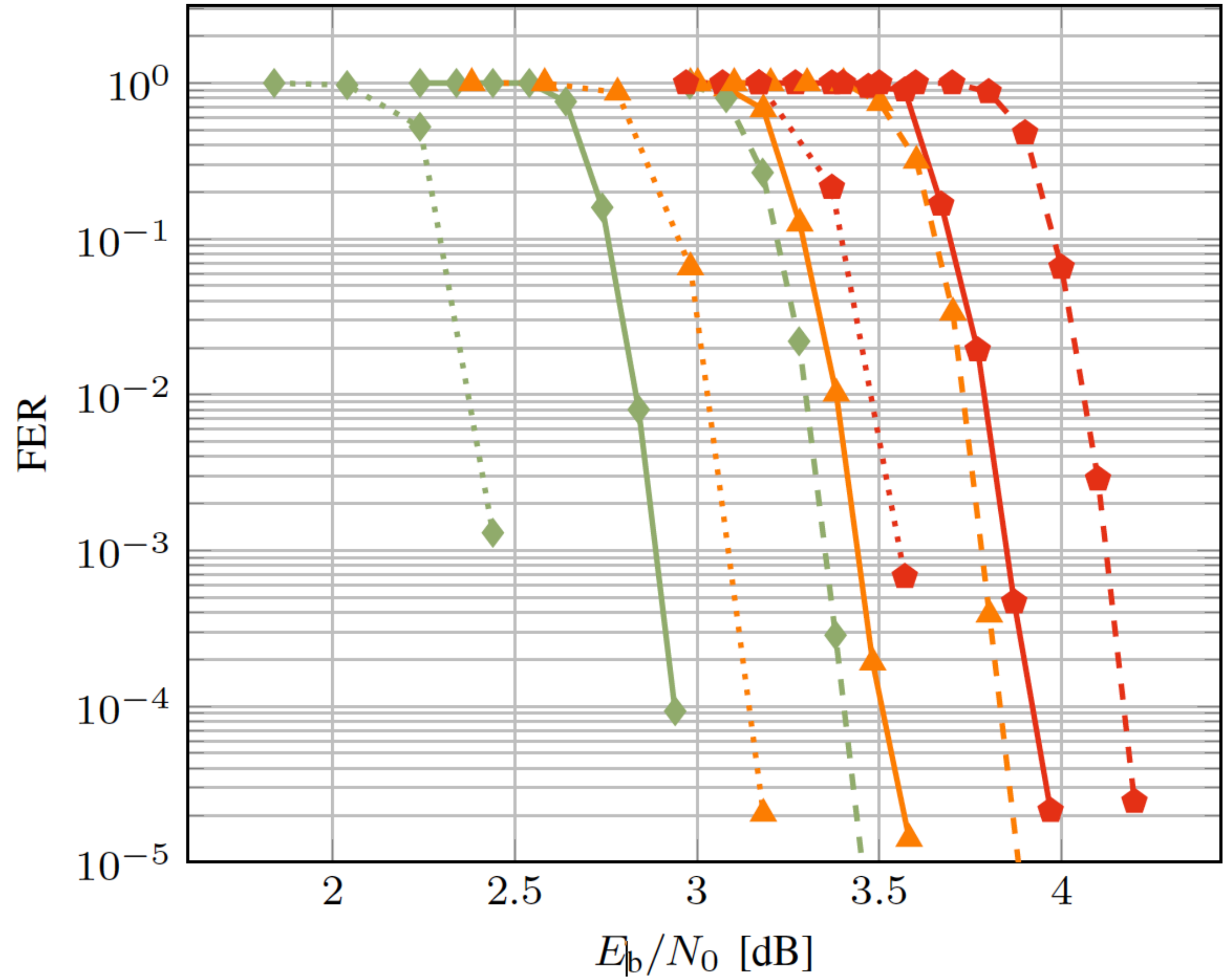


Fig. 2. FER versus E_b/N_0 for TMP and unquantized BP decoding for $R = 3/4$ (—), $R = 5/6$ (—) and $R = 7/8$ (—). We compare the TMP performance of optimized codes (—◆—, —▲—, —⬟—) to their AR4JA counterparts with unquantized BP (···◆···, ···▲···, ···⬟···) and TMP decoding (- -◆- -, - -▲- -, - -⬟- -).

Comments:

- Figure from E. Ben Yacoub et al.'s arxiv paper, Sep. 2018

- Curves show frame error rate (FER) of AR4JA and optimized codes

For More Details:

G. Lechner, T. Pedersen, and G. Kramer, “Analysis and design of binary message passing decoders,” IEEE Trans. Commun., 60(3), 601-607, 2012.
See also: http://arxiv.org/pdf/1004.4020v1

E. Ben Yacoub, “LDPC Decoding Algorithms Based on Ternary Message Passing,” Master’s Thesis, Technical University of Munich, Oct. 2018

E. Ben Yacoub, F. Steiner, B. Matuz, G. Liva, “Protograph-Based LDPC Code Design for Ternary Message Passing Decoding,” Sep. 2018
https://arxiv.org/abs/1809.10910v2

See the Posters!
And the First Talk Tomorrow!