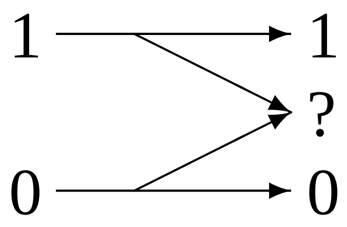
Code Annealing & the Suppressing Effect of the Cyclically Lifted LDPC Code Ensemble

Chih-Chun Wang

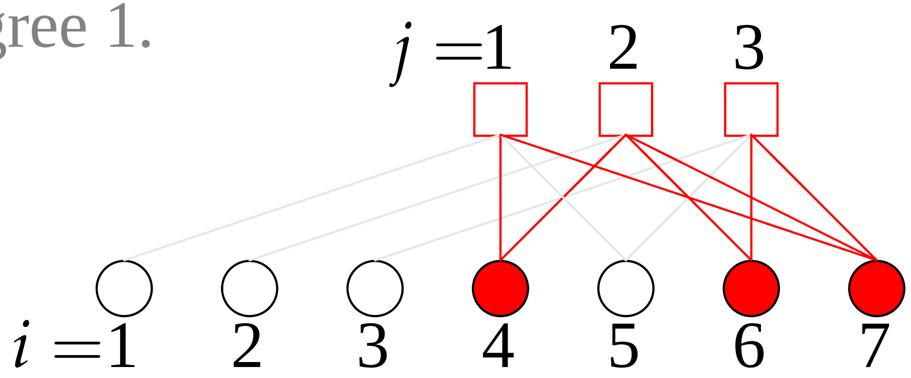
School of Electrical & Computer Engineering, Purdue Unversity

The Subject

- Binary erasure channels:



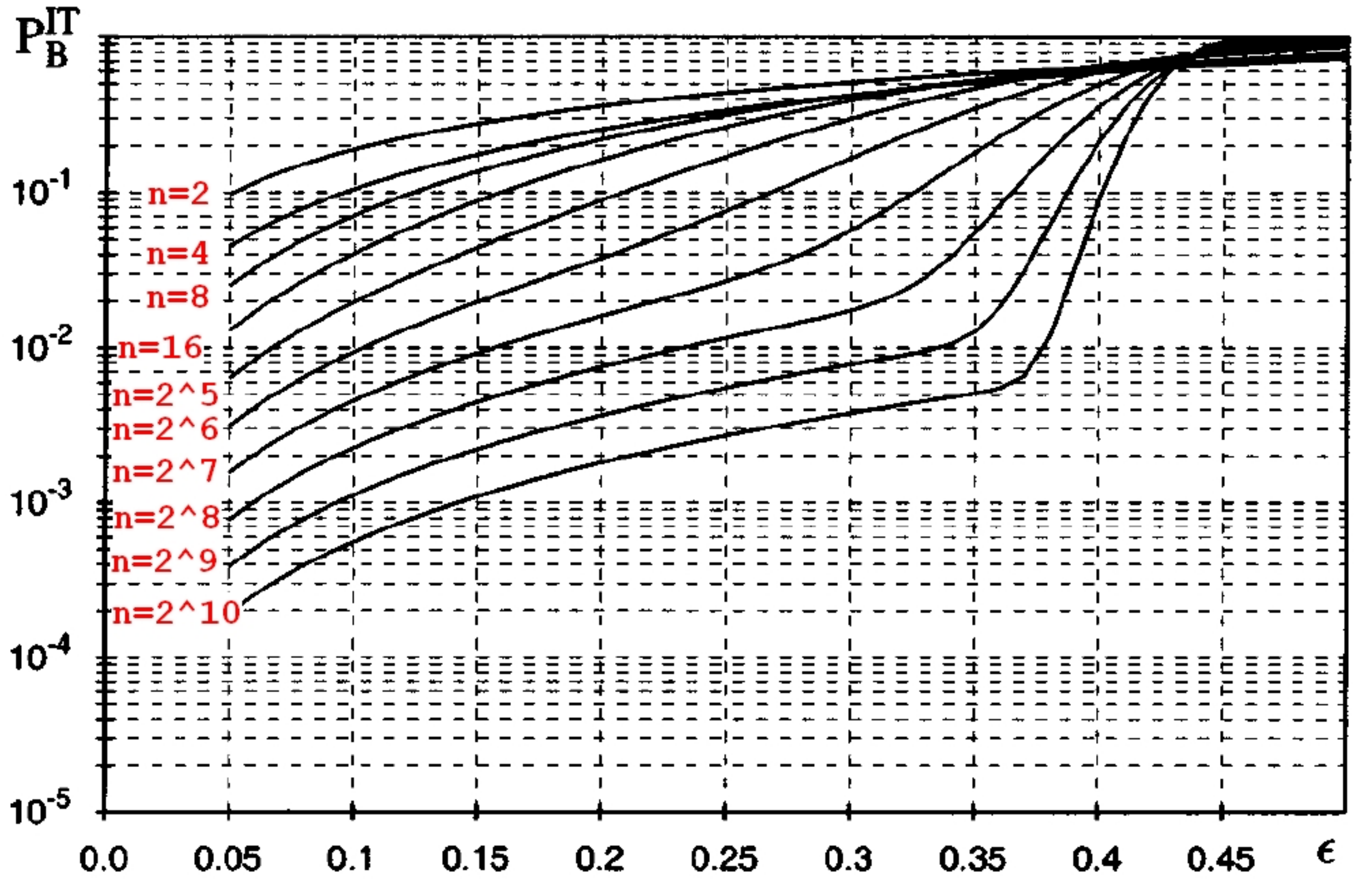
- Stopping Sets:
  - a set of variable nodes ⇒ the induced graph contains no check node of degree 1.



- Frame error rates
  - Frame decoding fails iff "the erasure bits" ⊇ "a stopping set."
- Error Floors ⟺ Minimal Stopping Distance.

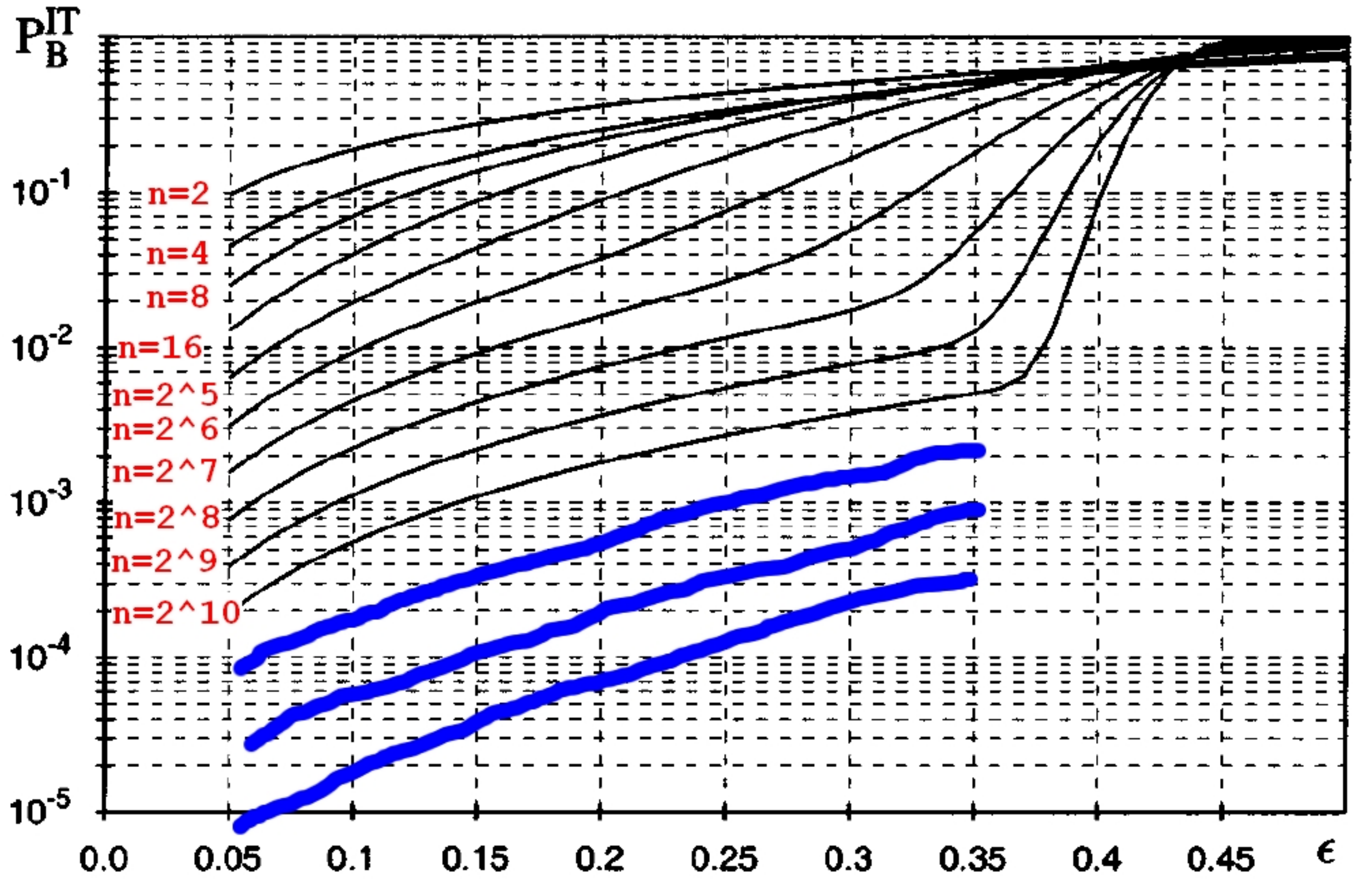
The Ensemble FER

[Di *et al.* 02] computes the ensemble FER by combinatorial methods.



The Ensemble FER

[Di *et al.* 02] computes the ensemble FER by combinatorial methods.

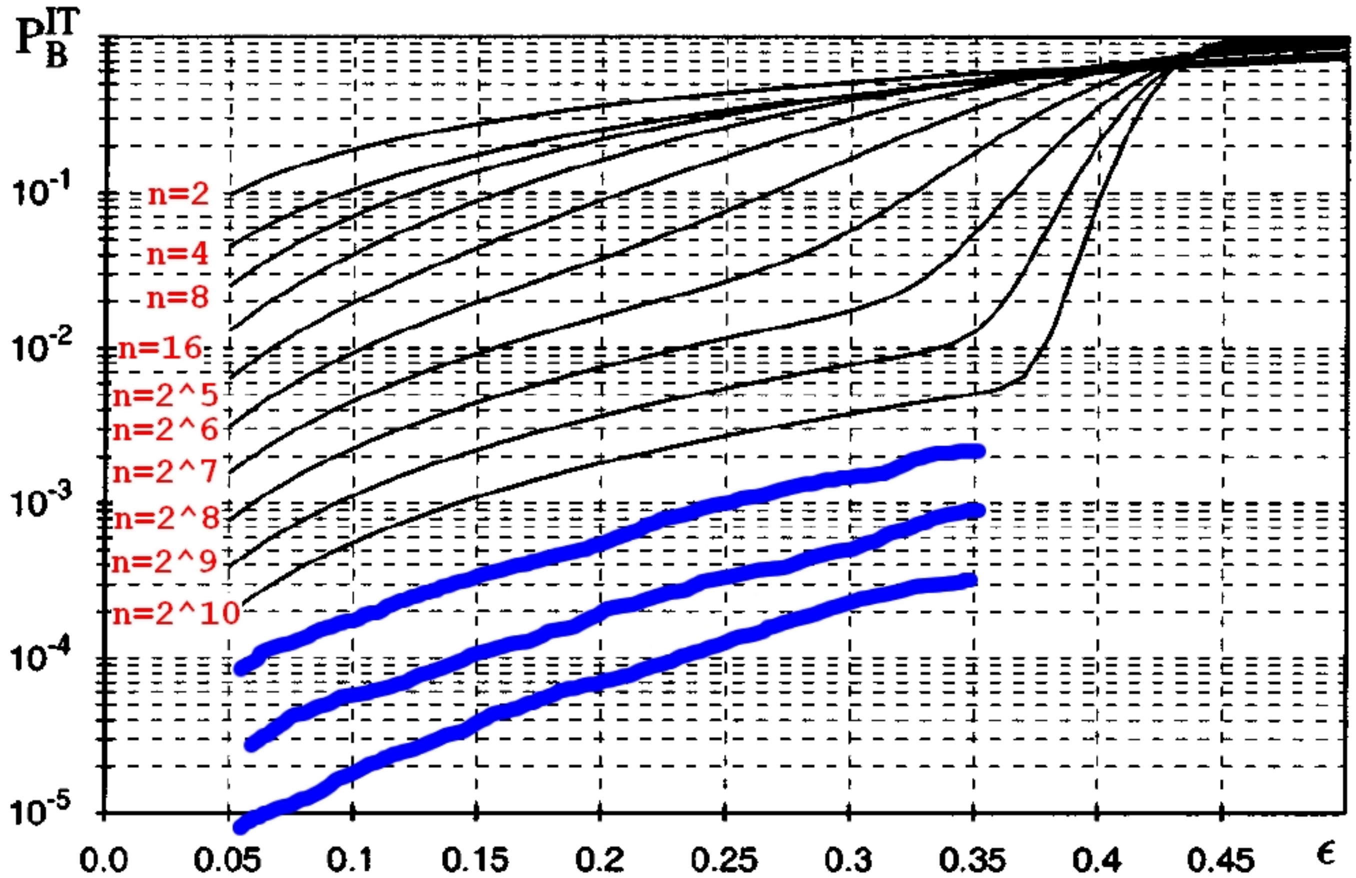


Error Floor Scaling Law?

[Montanari, Allerton 06] for bit-error rates

The Ensemble FER

[Di *et al.* 02] computes the ensemble FER by combinatorial methods.

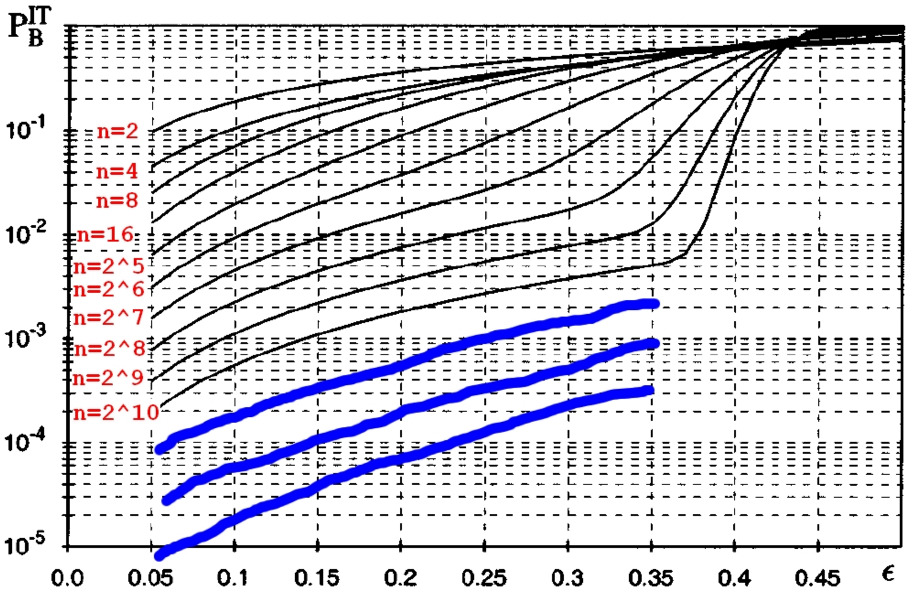


FER Error Floor Scaling Law?

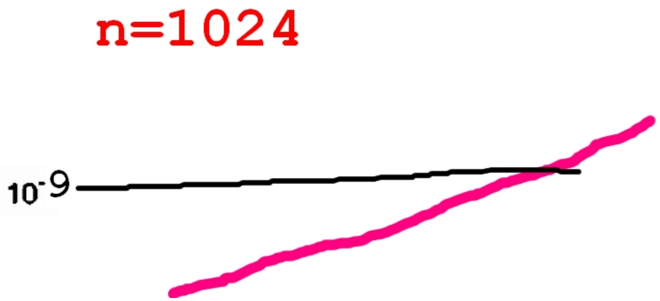
Irregular, Cyclic Lifted Ensemble?

The Ensemble FER

[Di *et al.* 02] computes the ensemble FER by combinatorial methods.

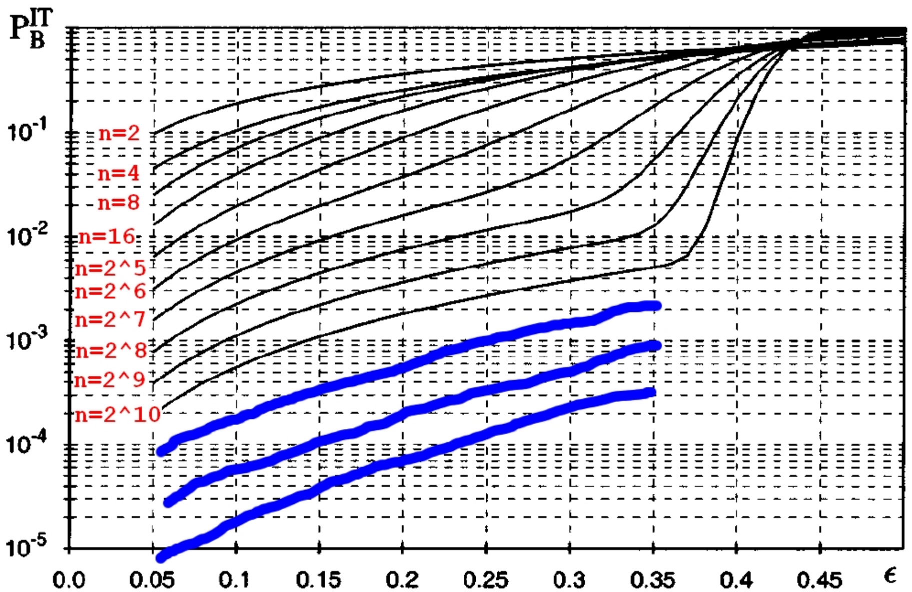


Non-constructive, expurgated short ensembles with very low FER error floor.

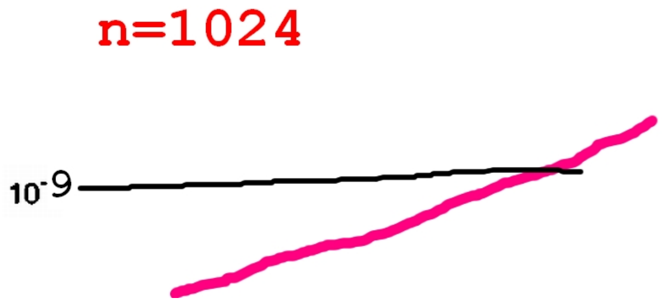


The Ensemble FER

[Di *et al.* 02] computes the ensemble FER by combinatorial methods.

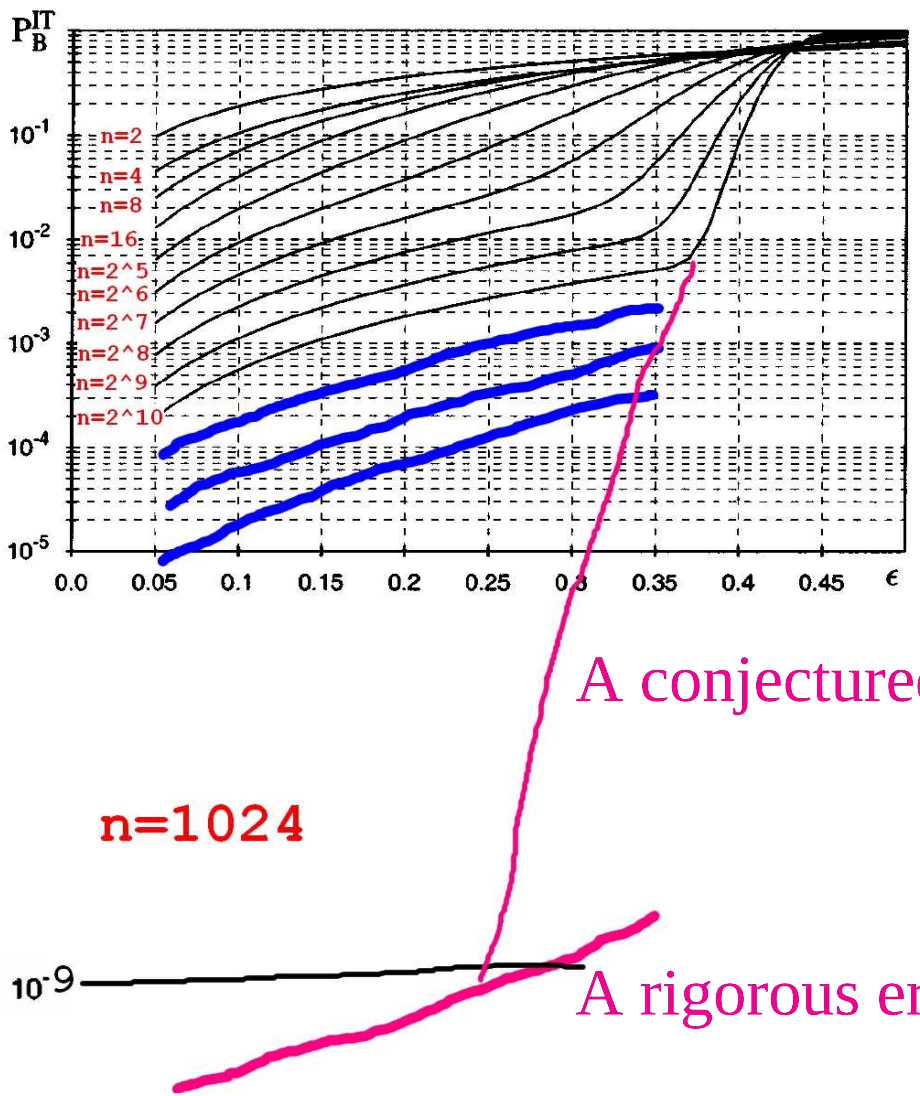


Constructive short ensembles with very low FER error floor.



The Ensemble FER

[Di *et al.* 02] computes the ensemble FER by combinatorial methods.



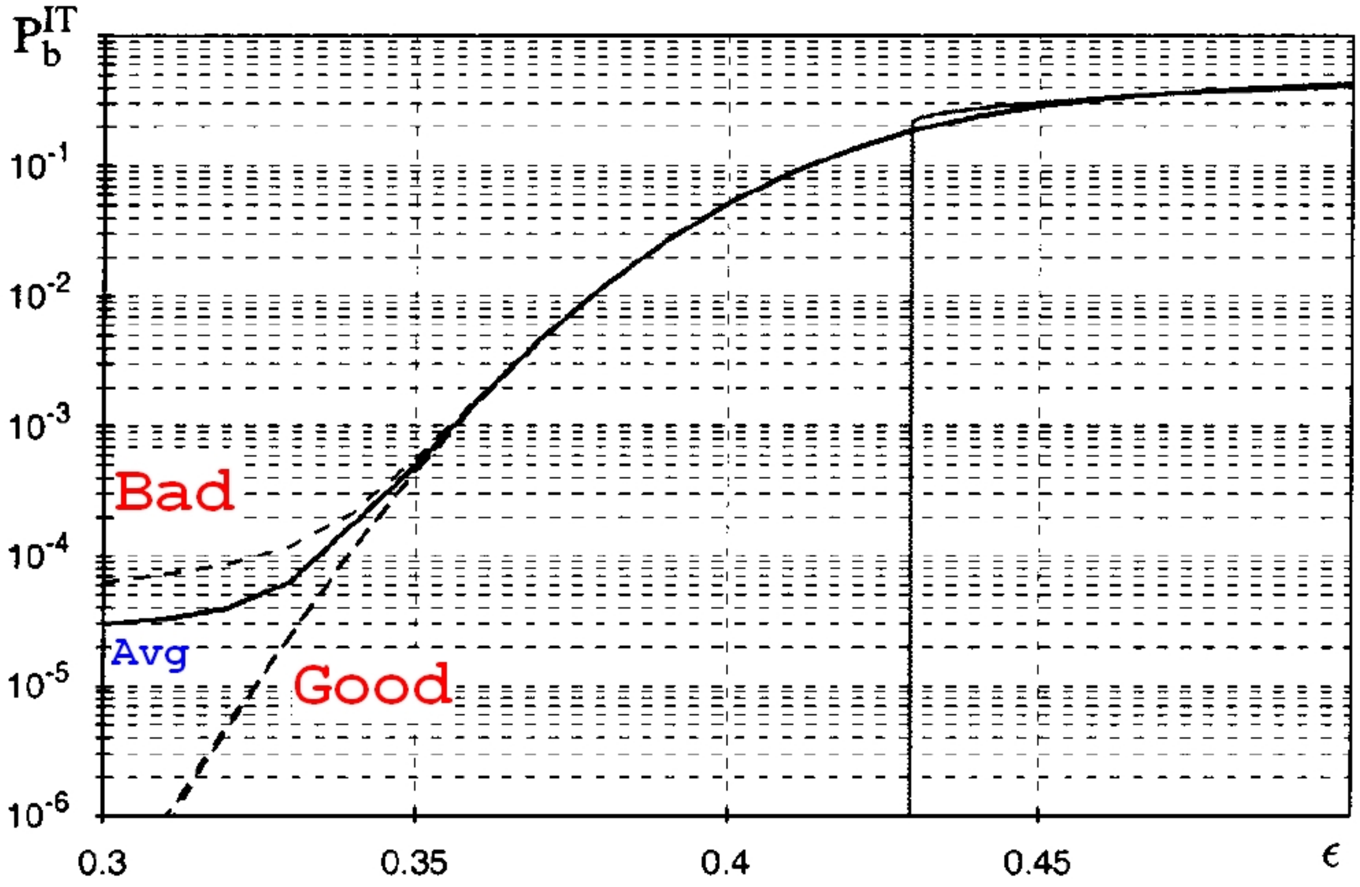
Constructive short ensembles with very low FER error floor.

A conjectured waterfall curve

A rigorous error floor curve

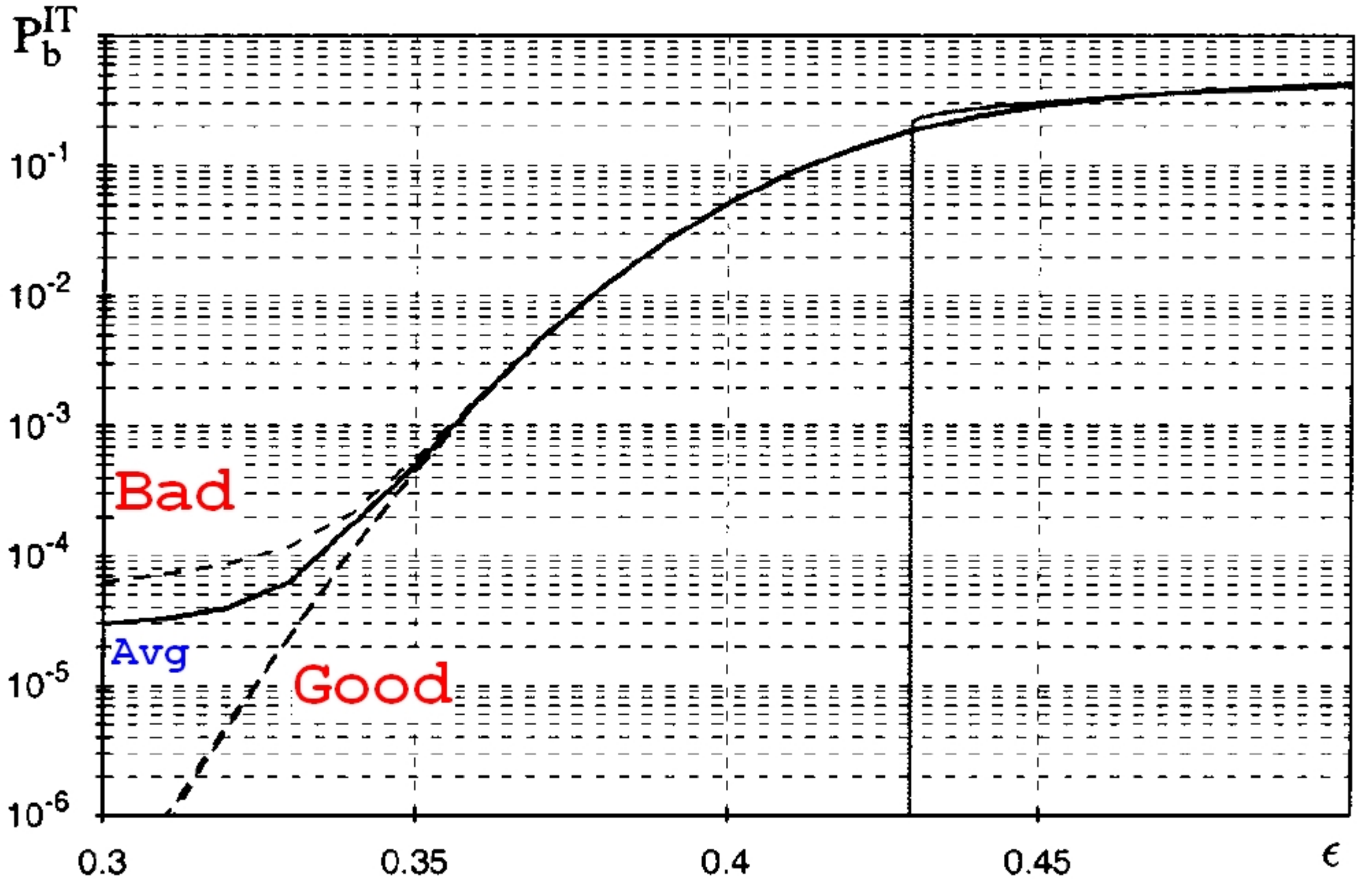
Finding Good Codes

Observation in [Di *et al.* 02].



Finding Good Codes

Observation in [Di *et al.* 02].



How to find good codes in a (good) ensemble?

Constructing Good Finite Codes

- Algebraic Approaches: [Kou *et al.* 01], [Vasic *et al.* 04]
- Progressive Edge Growth (PEG) constructions:
  - Girth — [Hu *et al.* 05],
  - Approximate Cyclic Extrinsic message degree — [Tian *et al.* 04],
  - Partial stopping set removal — [Ramamoorthy *et al.* 04],
  - Upper-bound-based construction — [Sharon *et al.* 06].
  - <u>Global construction</u> may change the <u>threshold</u> — [Sharon *et al.* 06].
- Loop Removal Construction: Girth — [McGowan *et al.* 03]

Constructing Good Finite Codes

- Algebraic Approaches: [Kou *et al.* 01], [Vasic *et al.* 04]
- Progressive Edge Growth (PEG) constructions:
  - Girth — [Hu *et al.* 05],
  - Approximate Cyclic Extrinsic message degree — [Tian *et al.* 04],
  - Partial stopping set removal — [Ramamoorthy *et al.* 04],
  - Upper-bound-based construction — [Sharon *et al.* 06].
  - Global construction may change the threshold — [Sharon *et al.* 06].
- Loop Removal Construction: Girth — [McGowan *et al.* 03]
- Code Annealing: Base on minimal stopping distance.

Stopping Set Exhaustion

- Hardness
  - Frame-wise minimal stopping distance $\longrightarrow$ NP-complete [Krishnan *et al.* 06],
  - Bit-wise minimal stopping distance $\longrightarrow$ NP-complete [Wang *et al.* 06]

Stopping Set Exhaustion

- Hardness
  - Frame-wise minimal stopping distance $\longrightarrow$ NP-complete [Krishnan *et al.* 06],
  - Bit-wise minimal stopping distance $\longrightarrow$ NP-complete [Wang *et al.* 06]
- INPUT: the target bit v_i,

  OUTPUT: an exhaustive list of min stopping sets containing v_i
  - A branch and bound approach [Wang *et al.* 06]

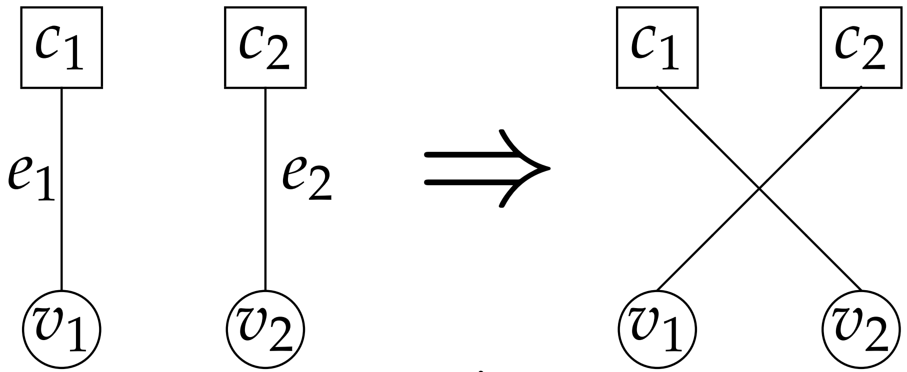
    Exhausting all stopping sets of size ≤ 13 for $n = 576$ codes.
  - The brute-force approach $\binom{576}{13} \approx 1.1 \times 10^{26}$.

Code Annealing Construction

Start with *any* code.

while time permits **do**

  Randomly select two edges e_1, e_2, and construct an *exhaustive list* of stopping sets $\mathcal{S}_{v_1,v_2}$

  Construct new $\mathcal{S}'_{v_1,v_2}$.

  if $\mathcal{S}'_{v_1,v_2} \prec \mathcal{S}_{v_1,v_2}$ **then**

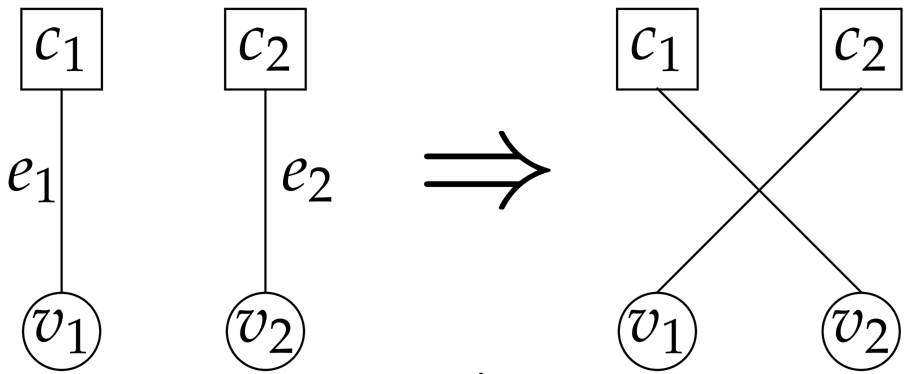
    Abandon the change.

  end if

end while

Code Annealing Construction

Start with *any* code.
while time permits **do**
  Randomly select two edges e_1, e_2, and construct an *exhaustive list* of stopping sets $\mathcal{S}_{v_1,v_2}$



  Construct new $\mathcal{S}'_{v_1,v_2}$.
  if $\mathcal{S}'_{v_1,v_2} \prec \mathcal{S}_{v_1,v_2}$ **then**
    Abandon the change.
  end if
end while

Features:
- Construct / polish codes
- Compatibility
- Local rearrangements
- No performance outlier

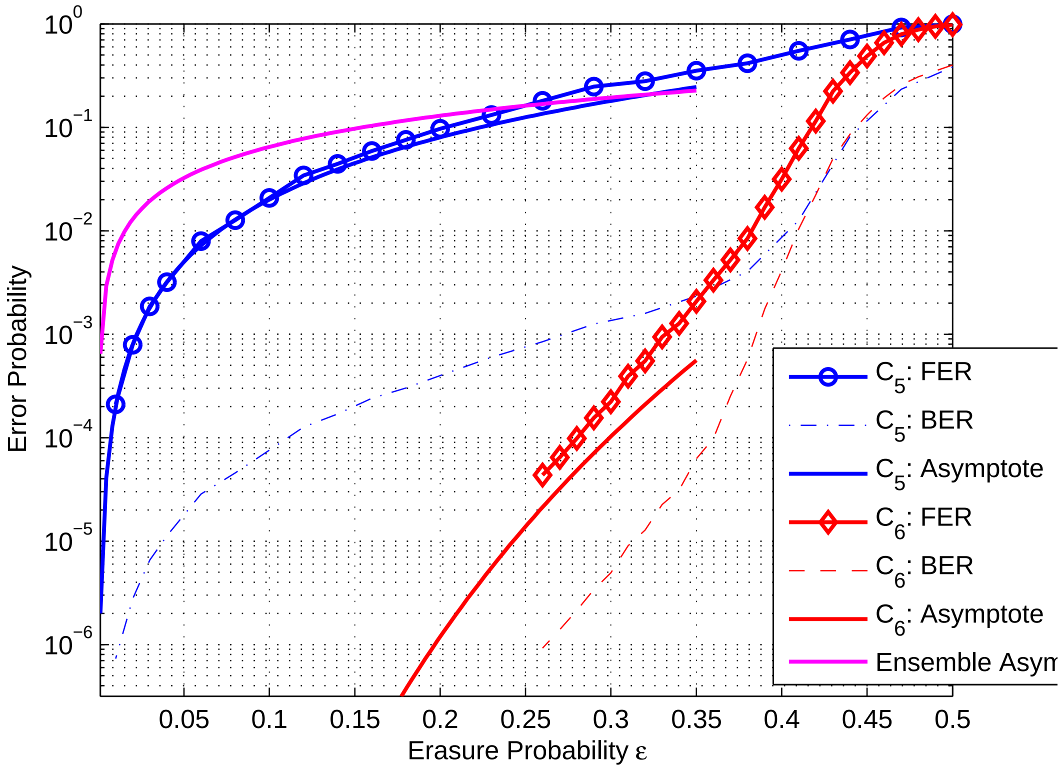
Performance

A $n = 576$, irregular rate $1/2$ code, rearranging 14.3% of edges.

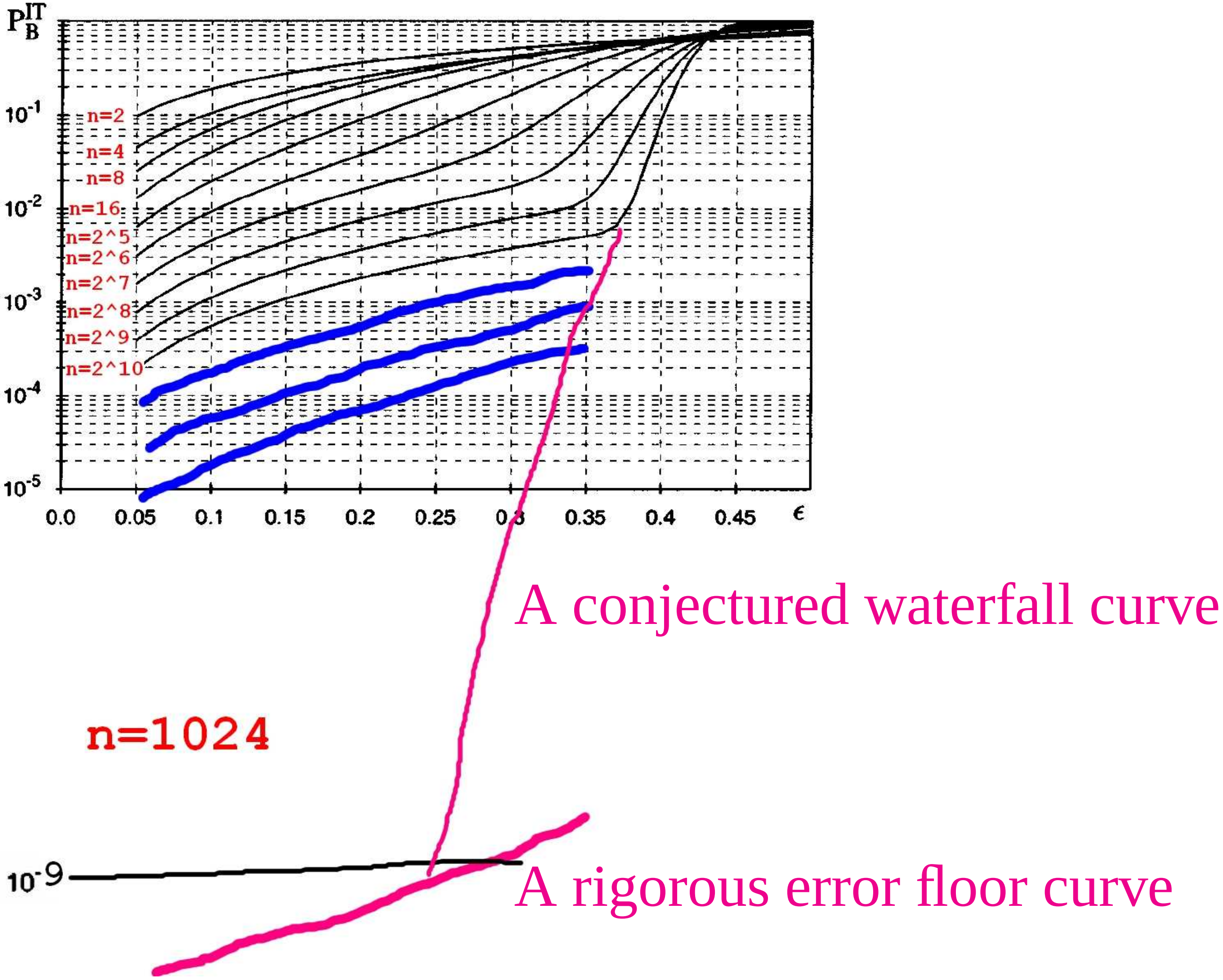
$$\lambda(x) = 0.419467x + 0.163384x^2 + 0.417149x^5,$$

$$\rho(x) = 0.002317x^3 + 0.997683x^5.$$

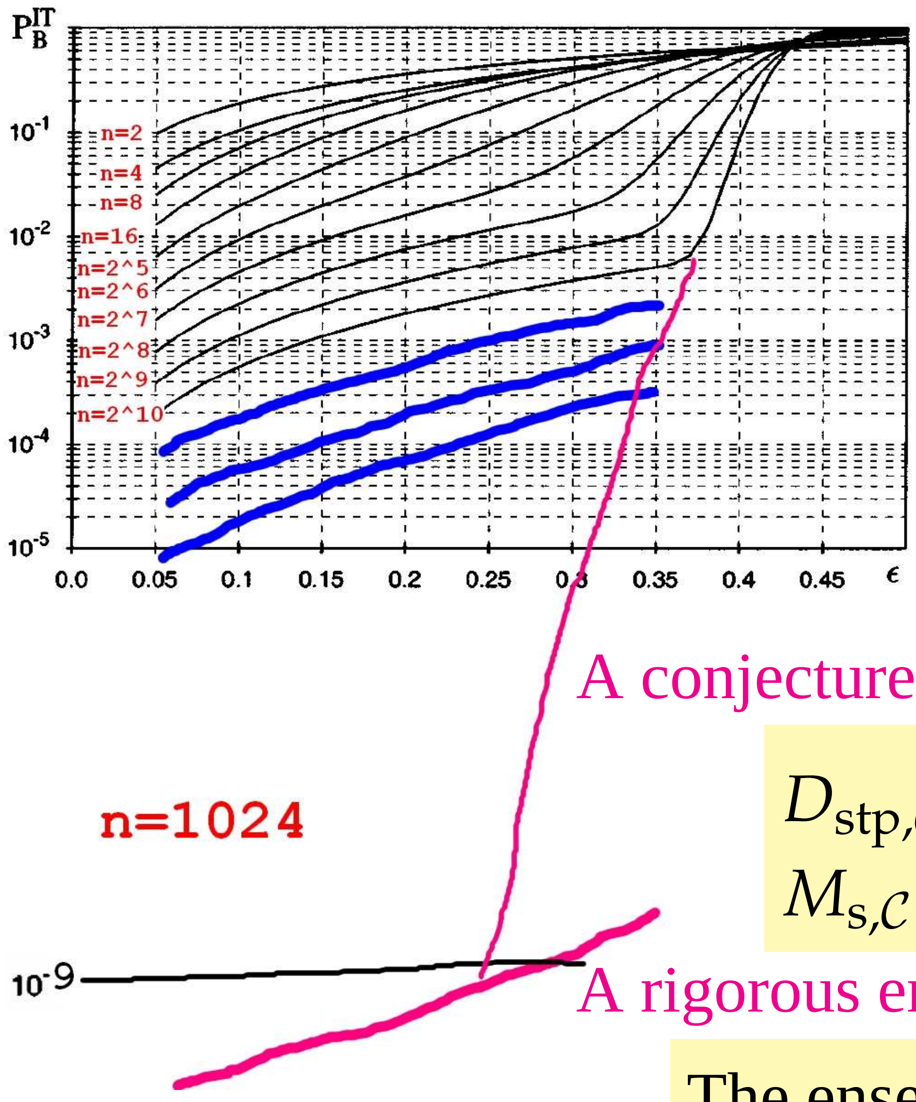
$(D_{\mathrm{stp}}, M_{\mathrm{s}})$: $(2,2) \to (11,58)$, top $\approx 0.12\%$ of the ensemble [Richardson *et al.* 02].



The Ensemble FER



The Ensemble FER



A conjectured waterfall curve

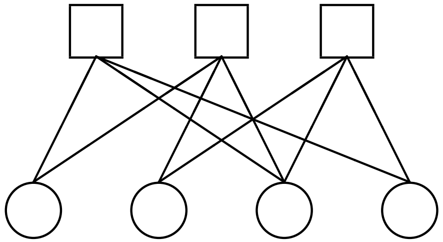
$D_{\text{stp},\mathcal{C}} \triangleq \min_{C \in \mathcal{C}} D_{\text{stp}}$,
$M_{\text{s},\mathcal{C}}$ be the expected multiplicity.

A rigorous error floor curve

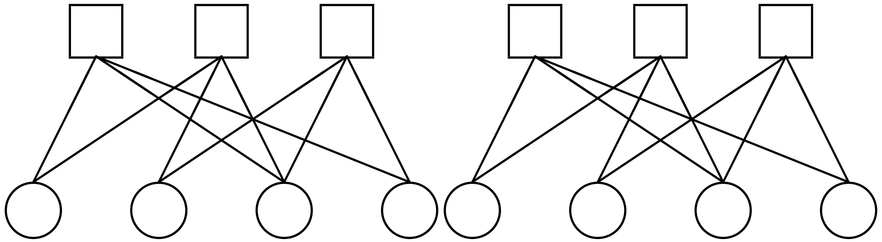
The ensemble error floor is $M_{\text{s},\mathcal{C}} \times \epsilon^{D_{\text{stp},\mathcal{C}}}$.

The Cyclically Lifted Ensemble

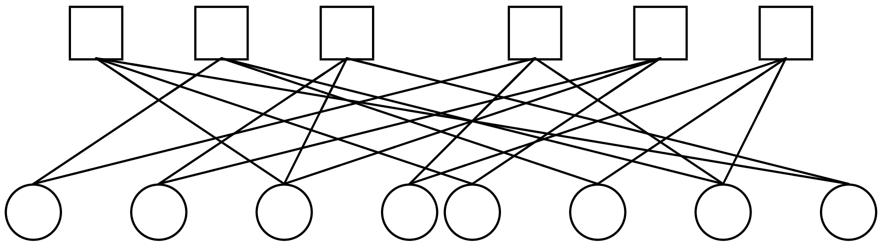
[Gross 74], [Richardson & Urbanke] and many more.



(a) The base code



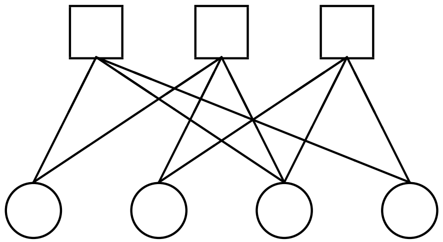
(b) The lifted code with an all-zero lifting sequence



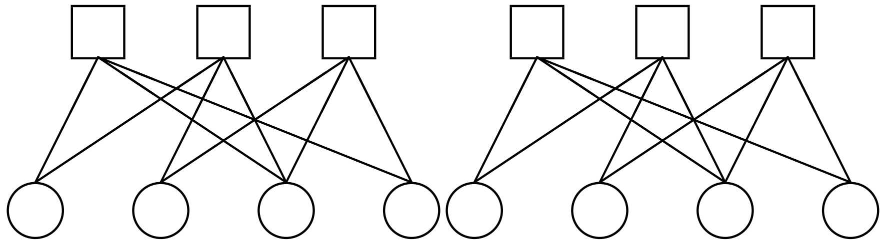
(c) The lifted code with a cyclic lifting sequence.

The Cyclically Lifted Ensemble

[Gross 74], [Richardson & Urbanke] and many more.

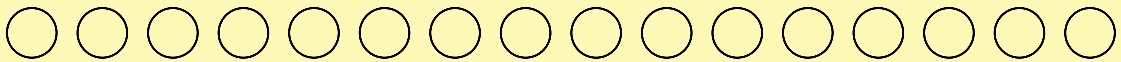


(a) The base code

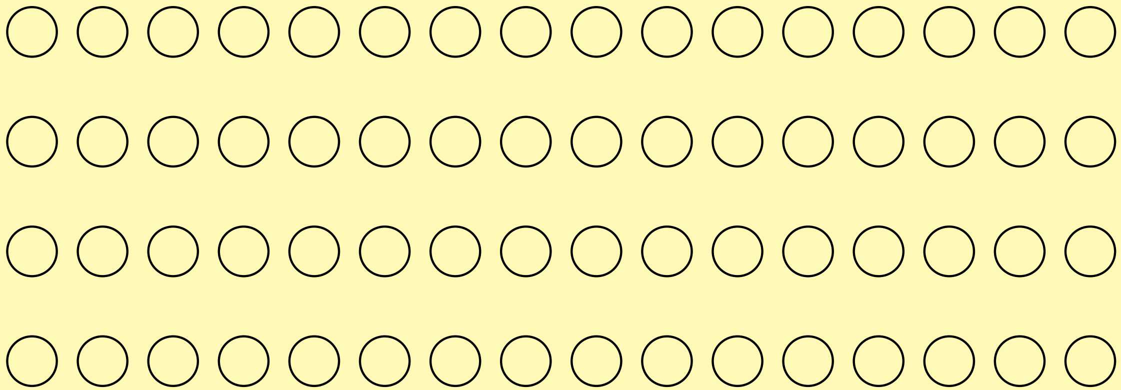


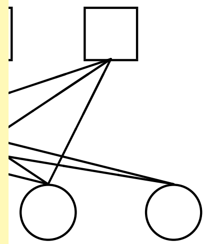
(b) The lifted code with an all-zero lifting sequence

Base Code — of size n ($n = 16$)



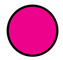
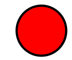
Lifted Code — of lifting factor K ($K = 4$)

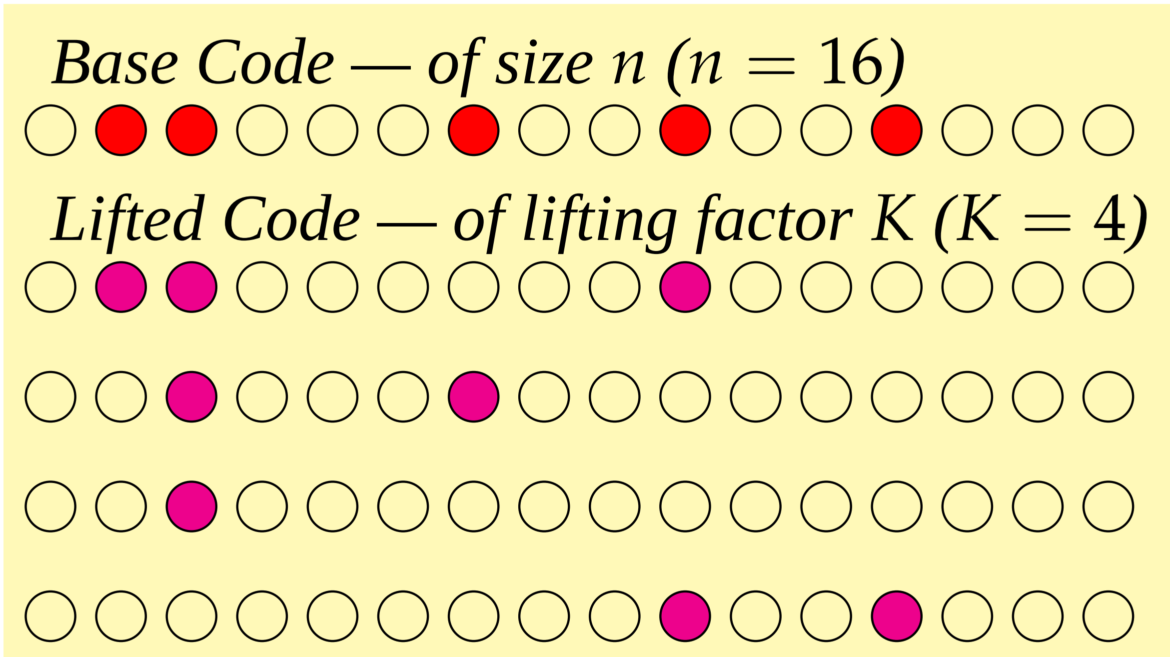




…fting sequence.

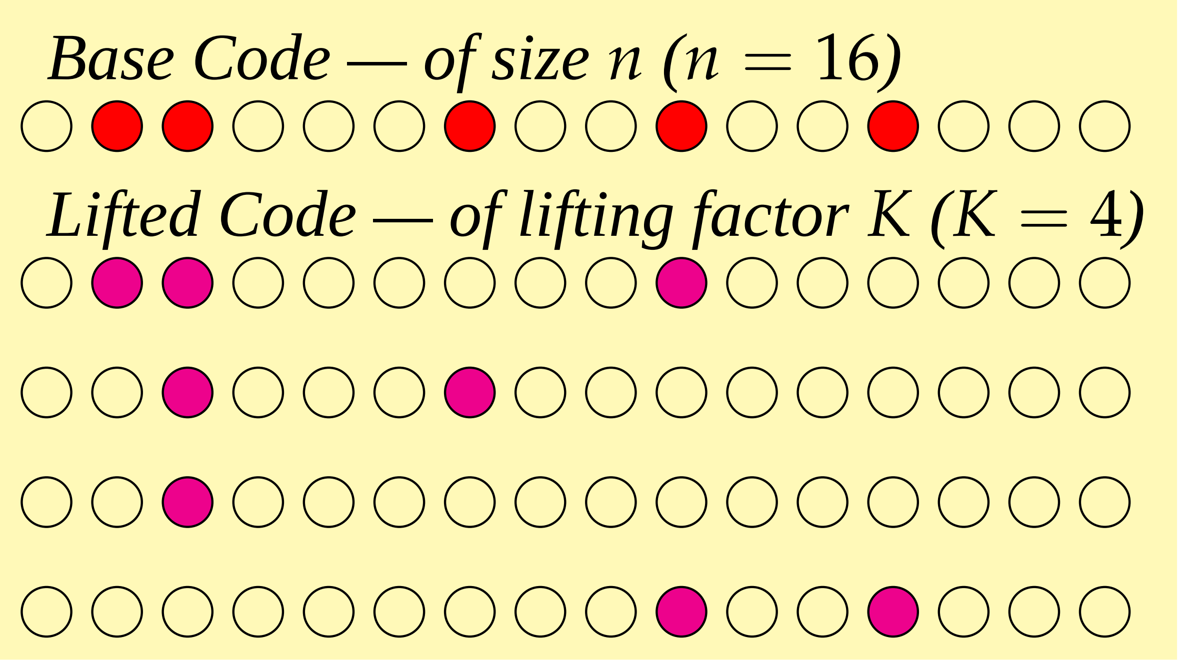
Determine $D_{\text{stp},\mathcal{C}}$

Theorem 1 *If*  *forms a stopping set for one lifted code, then*  *forms a stopping set for the base code.*



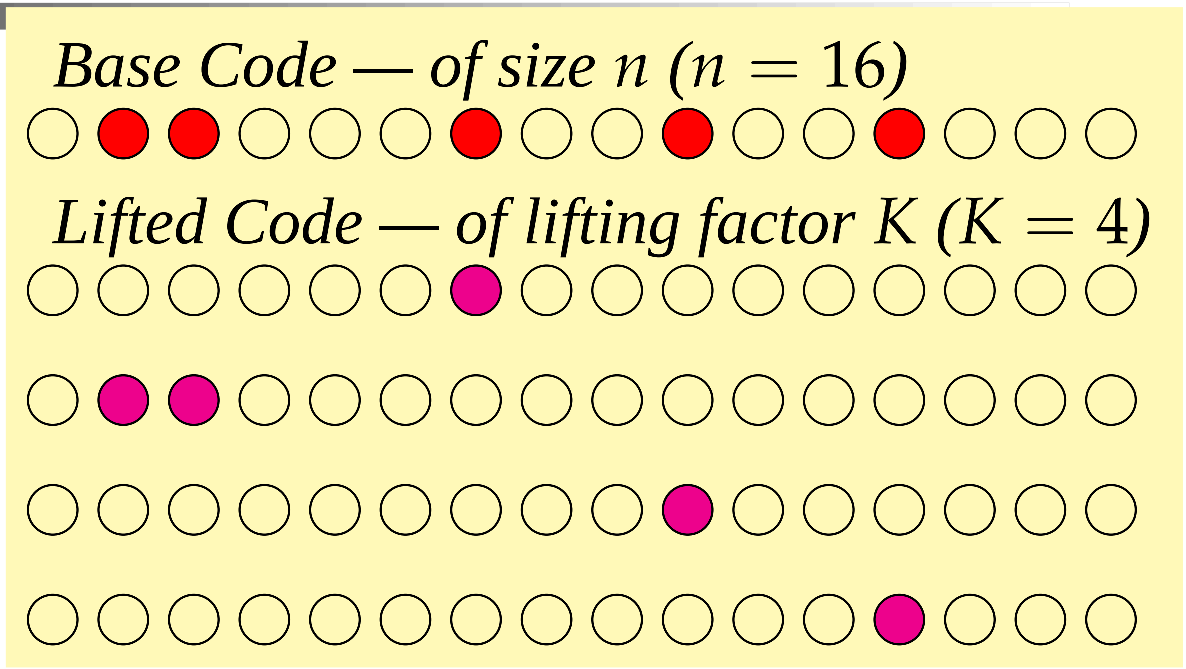
Determine $D_{\text{stp},\mathcal{C}}$

Theorem 1 *If* ● *forms a stopping set for one lifted code, then* ● *forms a stopping set for the base code.*



Corollary 1 $D_{\text{stp},\mathcal{C}}$ *equals* D_{stp} *of the base code.*

Different Order of Survivals



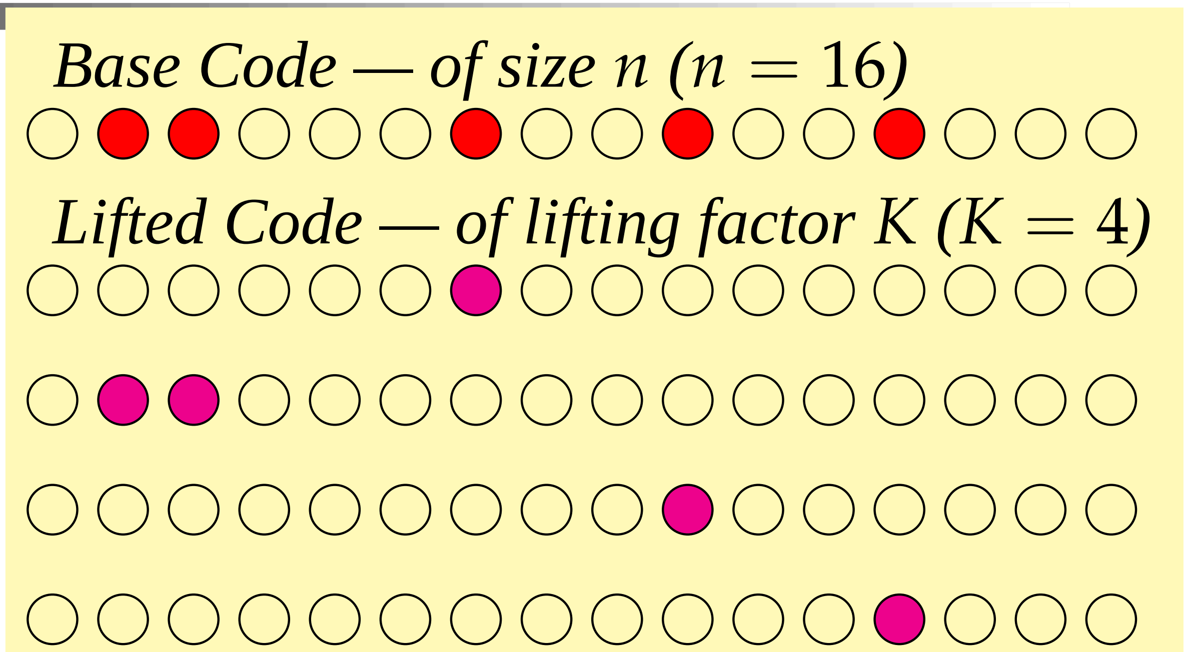
Definition 1

First order survivals

Different Order of Survivals

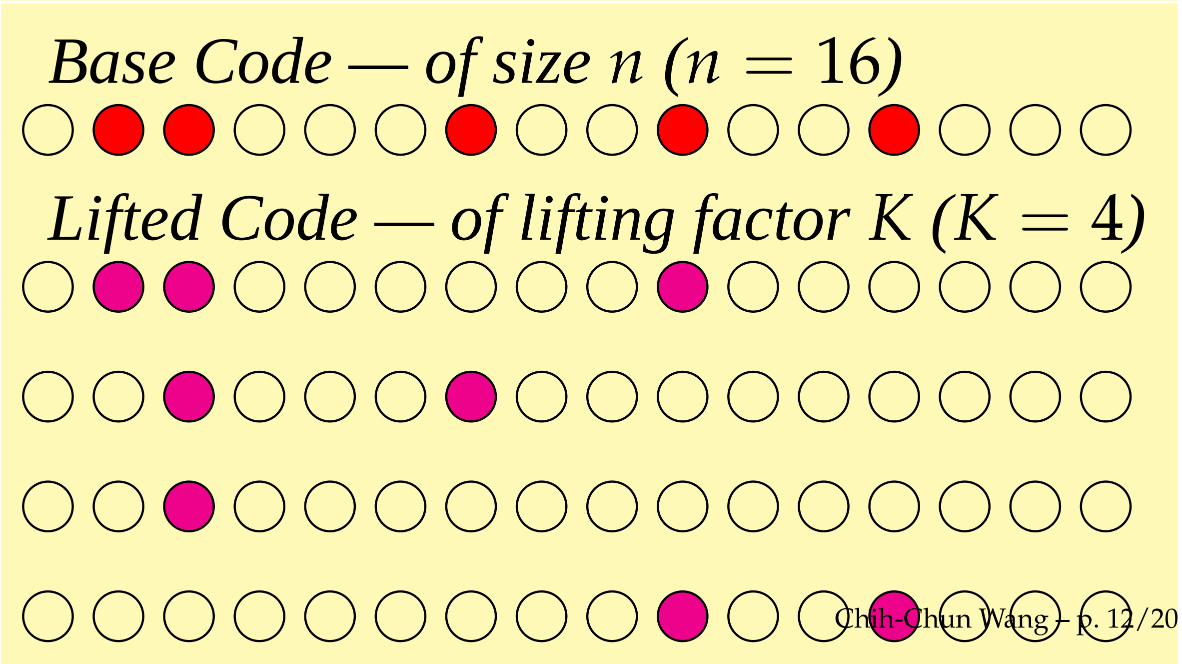
Definition 1

First order survivals



Definition 2

High order survivals



The 1st Order Suppressing Effect

Theorem 2 *For a fixed base code stopping set* $\mathbf{s}_B$*,*

$$\mathsf{E}\{|\textit{first order survivals}|\} =$$

$$K^{\#V-\#E}\prod_{j=1}^{\#C}\left(\sum_{t=0}^{\min(K,\deg(c_j))}(-1)^t\binom{\deg(c_j)}{t}\binom{K}{t}t!(K-t)^{\deg(c_j)-t}\right),$$

The 1st Order Suppressing Effect

Theorem 2 *For a fixed base code stopping set* $\mathbf{s}_B$,

$$\mathsf{E}\{|\textit{first order survivals}|\} =$$

$$K^{\#V-\#E}\prod_{j=1}^{\#C}\left(\sum_{t=0}^{\min(K,\deg(c_j))}(-1)^t\binom{\deg(c_j)}{t}\binom{K}{t}t!(K-t)^{\deg(c_j)-t}\right),$$

Theorem 3 (The Scaling Law of the Error Floor $M_{\mathrm{s},\mathcal{C}}\times\epsilon^{D_{\mathrm{stp},\mathcal{C}}}$**)**

$$\begin{aligned} D_{\mathrm{stp},\mathcal{C}} &= D_{\mathrm{stp}} \\ M_{\mathrm{s},\mathcal{C}} &= (1+o(1))\sum_{\textit{base min. stop. sets } \mathbf{s}_B} K^{-(0.5\#E-\#V+0.5\#C_{\mathrm{odd}})}, \end{aligned}$$

where $\#C_{\mathrm{odd}}$ *is the number of check nodes of odd degrees in the subgraph induced by* $\mathbf{s}_B$.

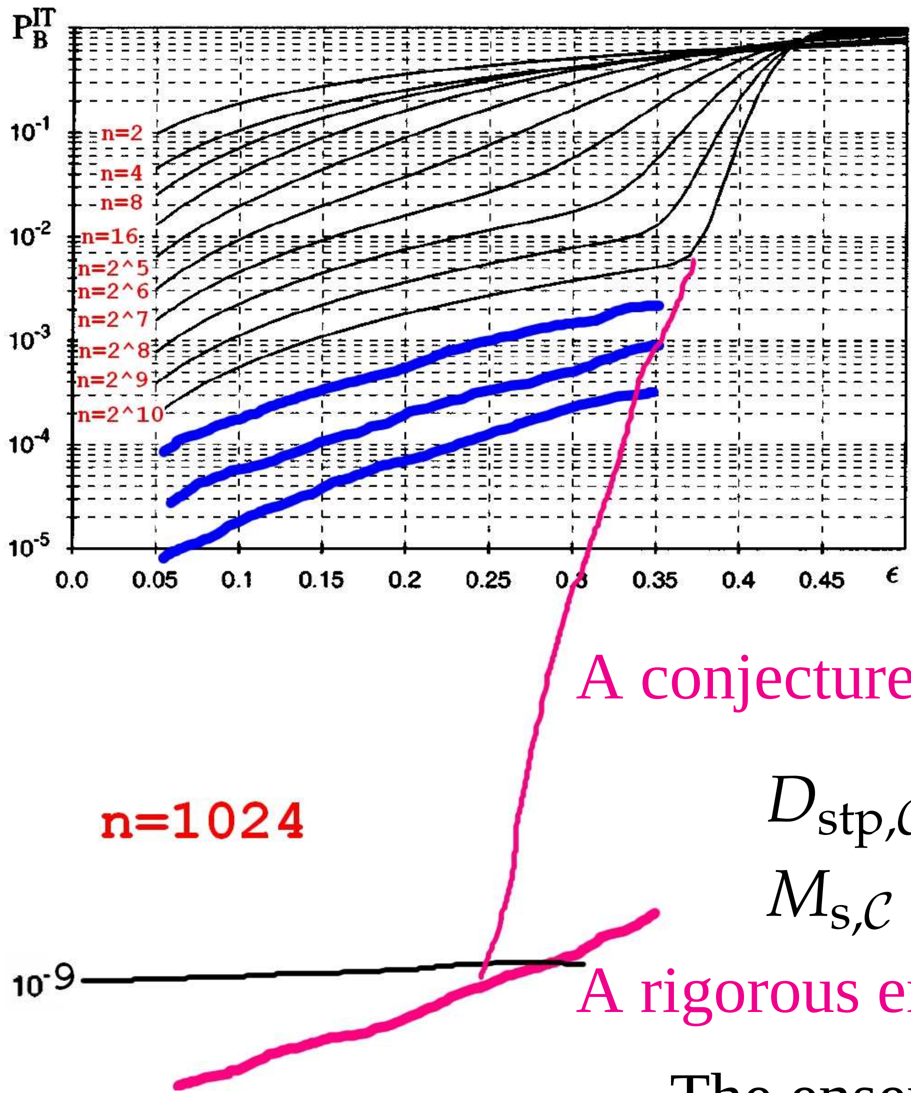
Construct Good Code Ensembles

- The error floor scaling law: $O\left(K^{-(0.5\#E-\#V+0.5\#C_{\text{odd}})}\right)$.
- Design criteria for the base code (neighborhood optimization):
  - Maximize D_{stp}.
  - Maximize the “minimal suppressing distance" for small stopping sets $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$.

Construct Good Code Ensembles

- The error floor scaling law: $O\left(K^{-(0.5\#E-\#V+0.5\#C_{\text{odd}})}\right)$.
- Design criteria for the base code (neighborhood optimization):
  - Maximize D_{stp}.
  - Maximize the "minimal suppressing distance" for small stopping sets
    $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$.
- For regular (3,6) codes:
  - $W_{\text{sup}} \geq 0.5 D_{\text{stp}}$
  - Optimize an ultra-short $n = 64$ (3,6) base code by code annealing $\Rightarrow D_{\text{stp}} = 8$. The suppressing effect is $1/65536$ for the $K = 16$ ($n = 1024$) lifted ensemble.

The (3,6) Code Ensemble FER



$$K^{-\left(0.5\#E-\#V+0.5\#C_{\text{odd}}\right)}$$

Double the codeword length n, the ensemble error floor can be lowered by 1–1.5 order of magnitude.

$D_{\text{stp},\mathcal{C}} \stackrel{\Delta}{=} \min_{C\in\mathcal{C}} D_{\text{stp}}$,

$M_{\text{s},\mathcal{C}}$ be the expected multiplicity.

The ensemble error floor is $M_{\text{s},\mathcal{C}} \times \epsilon^{D_{\text{stp},\mathcal{C}}}$.

High Order Suppressing Effects

Theorem 4 (An Algebraic Lower Bound)

$$\mathsf{E}\{|\textit{high order survivals}|\} = \begin{cases} 0 \\ \geq \text{const} \cdot \left(\max\left(K^{-(\#E_L - \#V_L - \#C_L)}, K^{-(\#E_B - \#V_B - \#C_B)}\right)\right) \end{cases}$$

Theorem 5 (An Algorithmic Upper Bound)

$$\mathsf{E}\{|\textit{high order survivals}|\} \leq \text{const} \cdot K^{\left(\sum_{v \in \mathbf{v}_{\text{o.d.}}} (R(v)-1)\right)} K^{-W_{\text{sup}}}.$$

Both bounds are tight for the first order suppressing effect.

High Order Suppressing Effects

Theorem 4 (An Algebraic Lower Bound)

$$\mathsf{E}\{|\textit{high order survivals}|\}$$

$$= \begin{cases} 0 \\ \geq \text{const} \cdot \left(\max\left(K^{-(\#E_L - \#V_L - \#C_L)}, K^{-(\#E_B - \#V_B - \#C_B)}\right)\right) \end{cases}$$

Theorem 5 (An Algorithmic Upper Bound)

$$\mathsf{E}\{|\textit{high order survivals}|\} \leq \text{const} \cdot K^{\left(\sum_{v \in \mathbf{v}_{\text{o.d.}}} (R(v)-1)\right)} K^{-W_{\text{sup}}}.$$

Both bounds are tight for the first order suppressing effect.

Why consider only the first order survivals?

Irregular Code Ensembles

- $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$
  - Degree 2 variable nodes are bad for cyclic lifting.
- Metric mismatch: For example, $n = 72$, $D_{\text{stp}} = 8$ but $W_{\text{sup}} = 0$.

Irregular Code Ensembles

- $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$
  - Degree 2 variable nodes are bad for cyclic lifting.
- Metric mismatch: For example, $n = 72$, $D_{\text{stp}} = 8$ but $W_{\text{sup}} = 0$.
- Base code optimization (neighborhood optimization): Degree augmentation + code annealing, which maximizes $D_{\text{stp}} + aW_{\text{sup}}$ for a predefined parameter a.

Irregular Code Ensembles

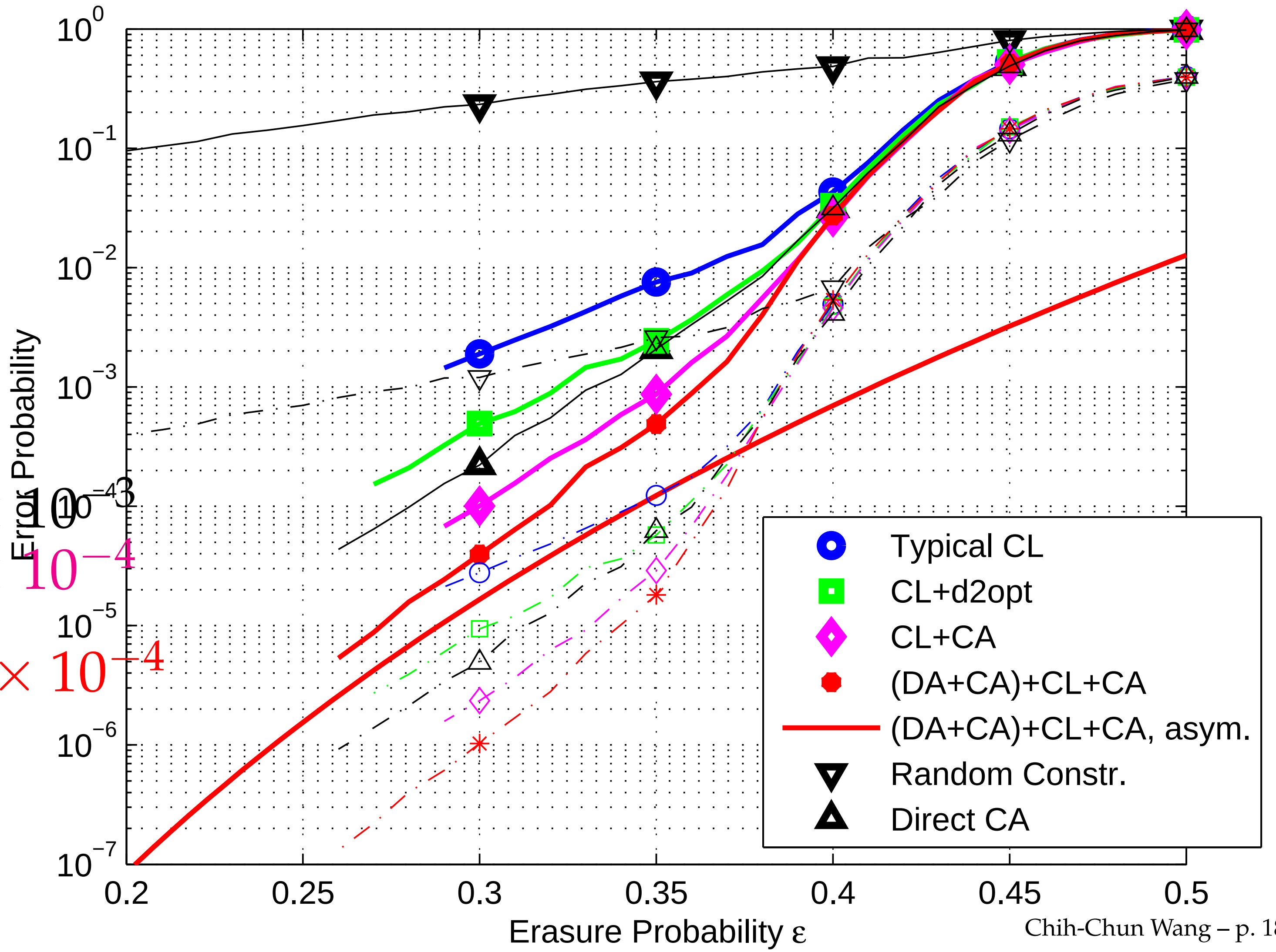
- $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$
  - Degree 2 variable nodes are bad for cyclic lifting.
- Metric mismatch: For example, $n = 72$, $D_{\text{stp}} = 8$ but $W_{\text{sup}} = 0$.
- Base code optimization (neighborhood optimization): Degree augmentation + code annealing, which maximizes $D_{\text{stp}} + aW_{\text{sup}}$ for a predefined parameter a.
- Cyclic lifting

Irregular Code Ensembles

- $W_{\mathrm{sup}} = 0.5\#E - \#V + 0.5\#C_{\mathrm{odd}}$
  - Degree 2 variable nodes are bad for cyclic lifting.
- Metric mismatch: For example, $n = 72$, $D_{\mathrm{stp}} = 8$ but $W_{\mathrm{sup}} = 0$.
- Base code optimization (neighborhood optimization): Degree augmentation + code annealing, which maximizes $D_{\mathrm{stp}} + aW_{\mathrm{sup}}$ for a predefined parameter a.
- Cyclic lifting
- Code annealing to remove those survival stopping sets of small sizes.

Performance

$$\lambda(x) = 0.416667x + 0.166667x^2 + 0.416667x^5, \rho(x) = x^5$$



(11,58), top 1.2×10^{-3}

(12,56), top 5.2×10^{-4}

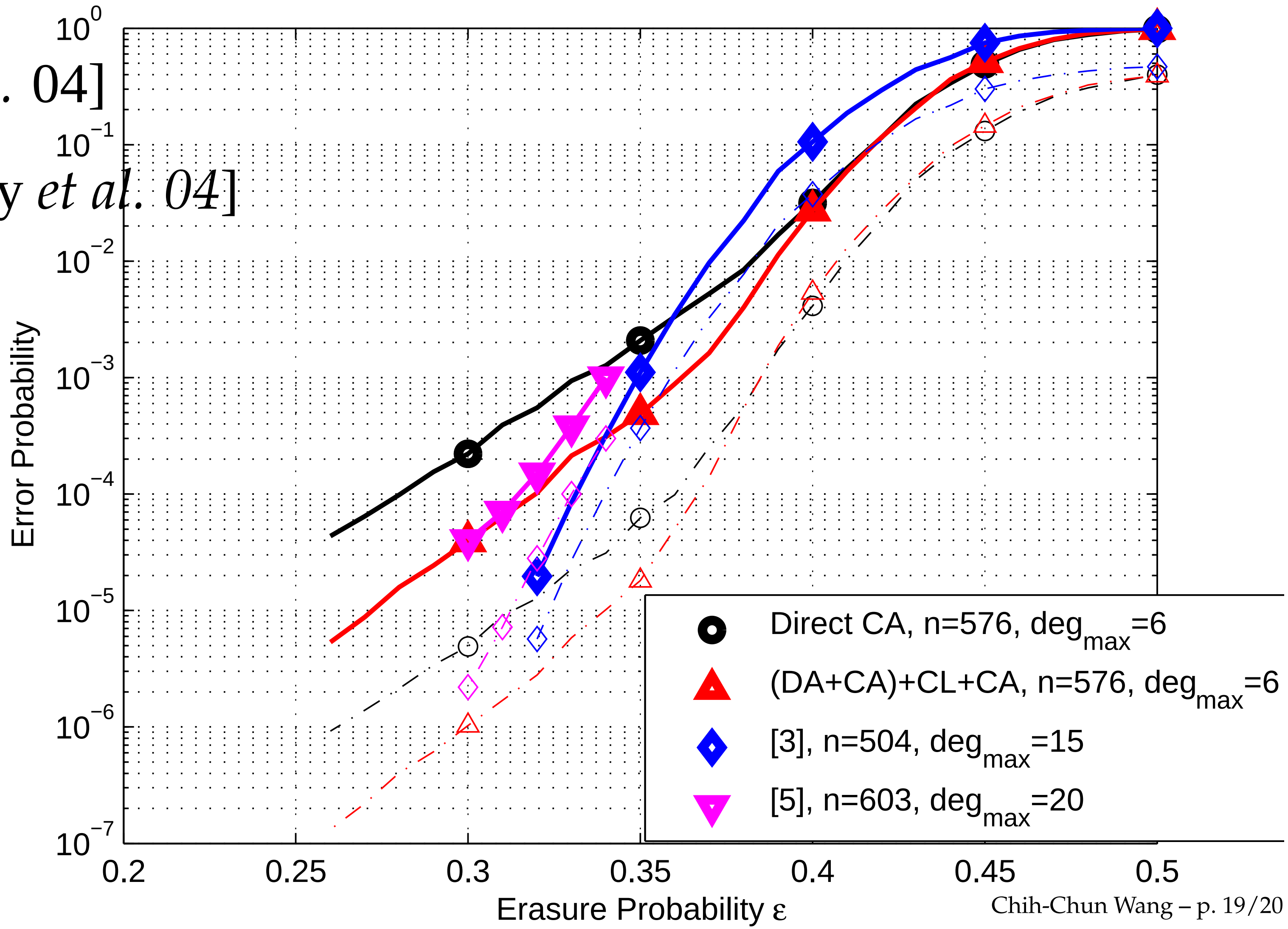
(13,104), top 2.2×10^{-4}

Performance

$$\lambda(x) = 0.416667x + 0.166667x^2 + 0.416667x^5,\ \rho(x) = x^5$$

[3] PEG [Hu *et al.* 04]

[5] [Ramamoorthy *et al.* 04]



Performance

$$\lambda(x) = 0.416667x + 0.166667x^2 + 0.416667x^5, \rho(x) = x^5$$

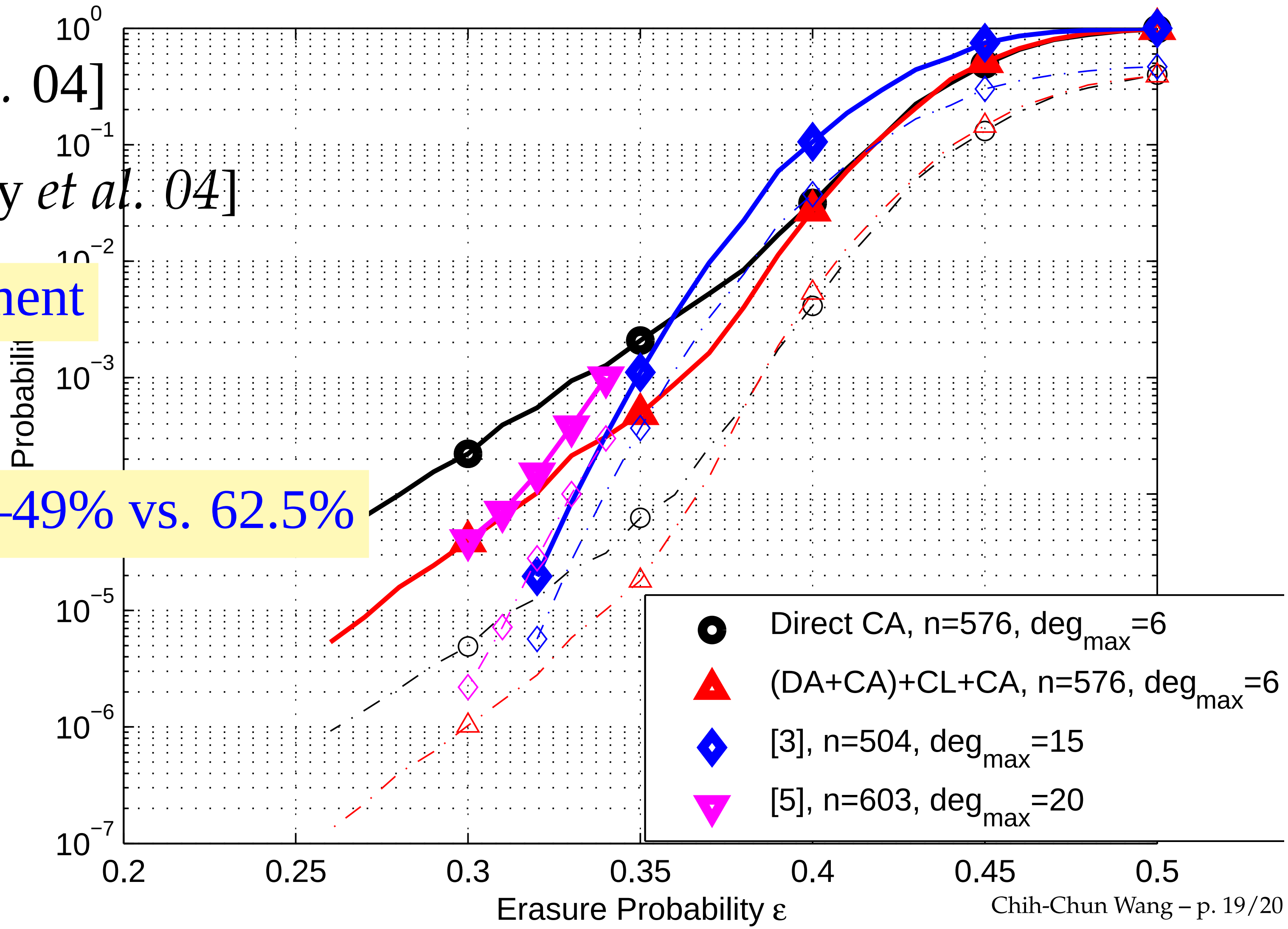
[3] PEG [Hu *et al.* 04]

[5] [Ramamoorthy *et al.* 04]

Local rearrangement

small $\deg_{max}$

$\deg(v) = 2$: 47–49% vs. 62.5%



Summary

- Code annealing
  - Exploiting the new tool of stopping set exhaustion.
  - Construct and polish good codes.
- Suppressing effects
  - Quantifying different orders survivals after cyclic lifting.
  - Implying an ensemble FER error floor scaling law.
  - Neighborhood optimization based on $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$.
- Constructing good irregular codes $\sim$ by degree augmentation + CA + CL + CA

Summary

- Code annealing
  - Exploiting the new tool of stopping set exhaustion.
  - Construct and polish good codes.
- Suppressing effects
  - Quantifying different orders survivals after cyclic lifting.
  - Implying an ensemble FER error floor scaling law.
  - Neighborhood optimization based on $W_{\text{sup}} = 0.5\#E - \#V + 0.5\#C_{\text{odd}}$.
- Constructing good irregular codes $\sim$ by degree augmentation + CA + CL + CA
- Non-erasure channels?