Decoding Turbo Codes and LDPC Codes via Linear Programming

Jon Feldman

jonfeld@theory.lcs.mit.edu

MIT LCS

David Karger

karger@theory.lcs.mit.edu

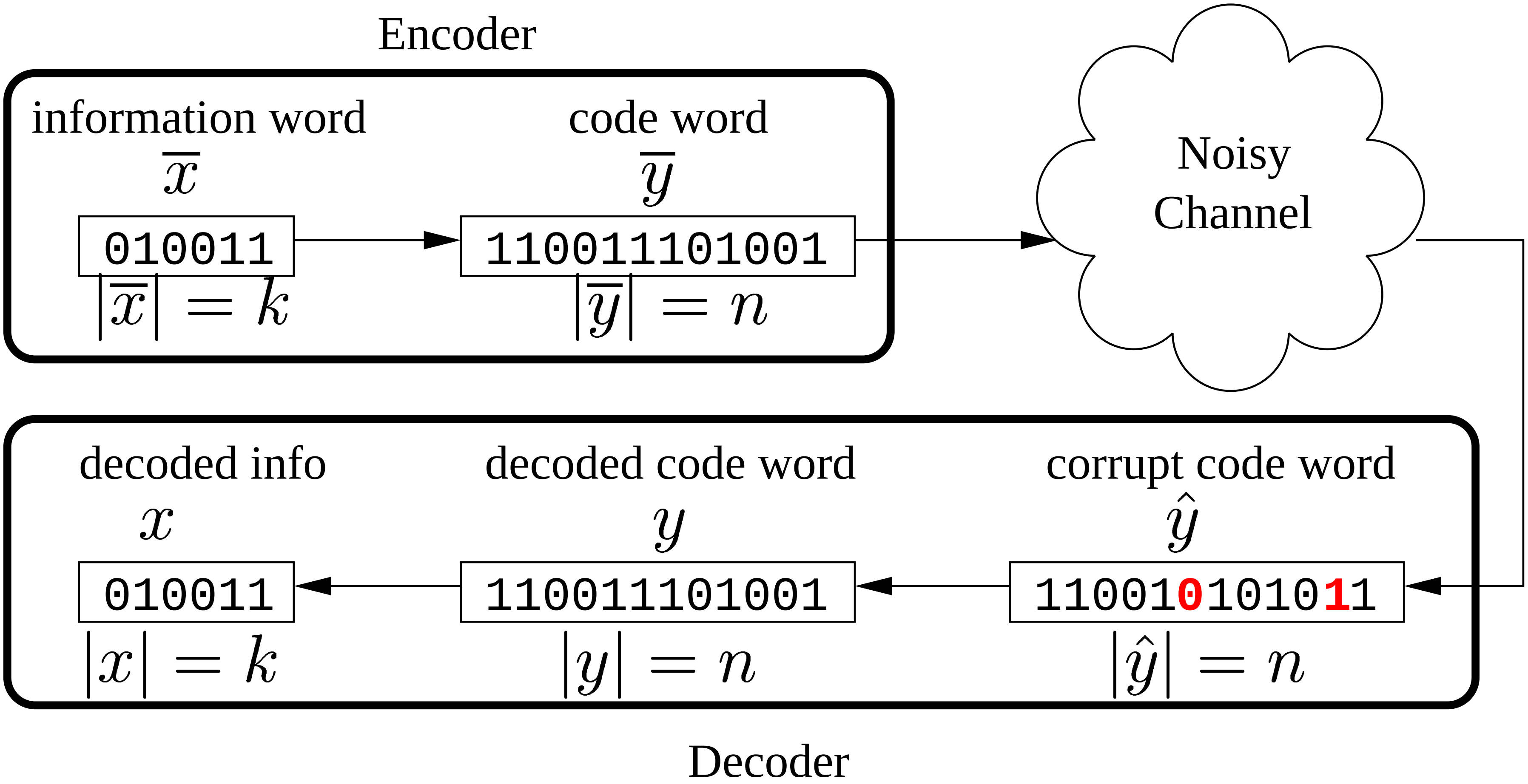
MIT LCS

Martin Wainwright

martinw@eecs.berkeley.edu

UC Berkeley

Binary Error-Correcting Code



- Binary Symmetric Channel (BSC): each bit flipped independently with probability p (small constant).

Turbo Codes + LDPC Codes

- Low-Density Parity-Check (LDPC) codes [Gal ’62] .
- Turbo Codes introduced [BGT ’93], unprecedented error-correcting performance.
- Ensuing LDPC “Renaissance” [SS ’94, MN ’95, Wib ’96, MMC ’98, Yed ’02, ...].
- Simple encoder, “belief-propagation” decoder.
- Theoretical understanding of good performance:
  - “Threshold” as $n \to \infty$ [LMSS ’01, RU ’01];
  - Decoder unpredictable with cycles.
- Finite-length analysis: combinatorial error conditions known only for the binary erasure channel [DPRTU ’02].

Our contributions

[FK, FOCS ’02] [FKW, Allerton ’02] [FKW, CISS ’03]

- Poly-time decoder using LP relaxation.
- Decodes: binary linear codes $\supseteq$ LDPC codes $\supseteq$ turbo codes.
- “Pseudocodewords:” exact characterization of error patterns causing failure.
- “Fractional distance” δ:
  - LP decoding corrects up to $\delta/2$ errors.
  - Computable efficiently for turbo, LDPC codes.
- Error rate bounds based on high-girth graphs.
- Closely related to iterative approaches, other notions of “pseudocodewords.”

Outline

- Error correcting codes.
- Using LP relaxation for decoding.
- Details of LP relaxation for binary linear codes.
- Pseudocodewords.
- Fractional Distance.
- Girth-based bounds.

Maximum-Likelihood Decoding

- Code $C \subset \{0,1\}^n$.
- Cost function γ_i: negative log-likelihood ratio of y_i.
- BSC: $\gamma_i = +1$ if $\hat{y}_i = 0$, $\gamma_i = -1$ if $\hat{y}_i = 1$.
- Other channels: γ_i takes on arbitrary "soft values."

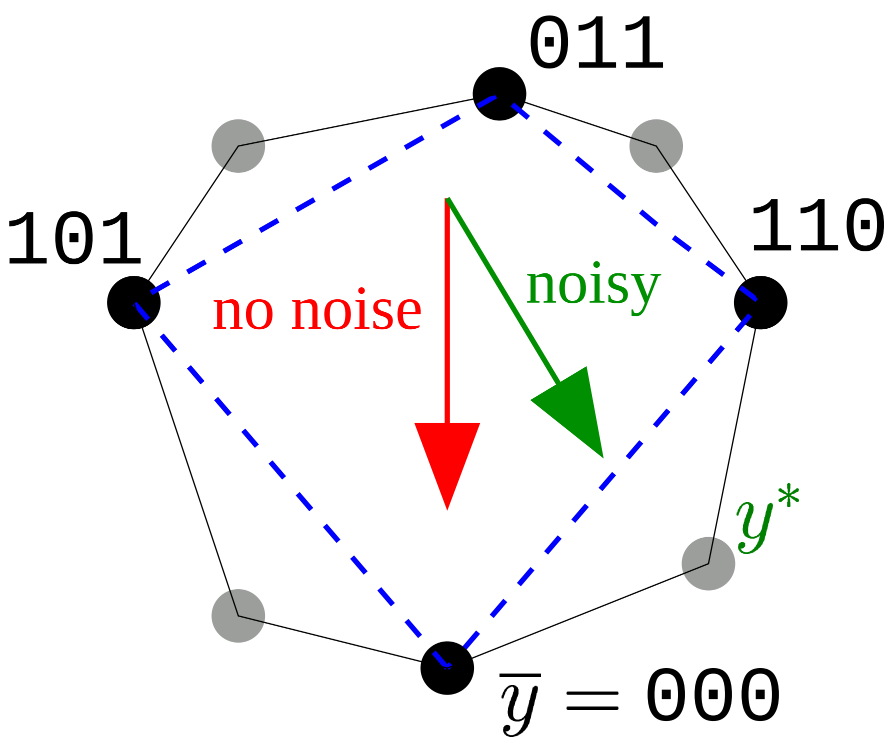
Given: Corrupt code word $\hat{y}$.
Find: $y \in C$ such that $\sum_i \gamma_i y_i$ is minimized.

- Linear Programming formulation:
  - Variables y_i for each code bit, $0 \le y_i \le 1$.
  - Linear Program:

$$\text{Minimize } \sum_i \gamma_i y_i \text{ s.t. } y \in \mathrm{CH}(C).$$

Linear Programming Relaxation

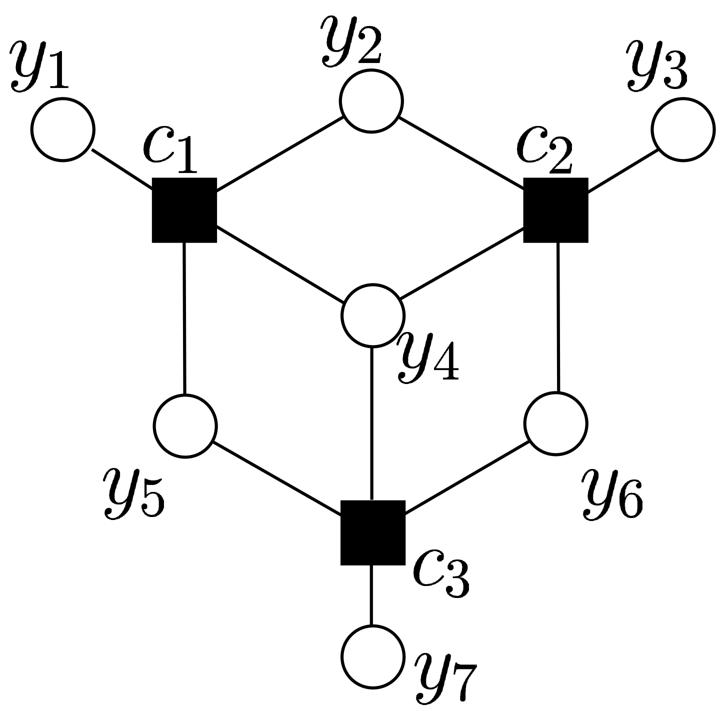
- Polytope P: relaxation, $C = P \cap \{0,1\}^n$.
- Decoder: Solve LP using simplex/ellipsoid. If $y^* \in \{0,1\}^n$, output y^*, else output "error."
- *ML certificate* property: all outputs ML codewords.
- Want low word error rate (WER) := $\Pr_{\text{noise}}[\ y \neq \overline{y}\ ]$.



- Min $\sum_i \gamma_i y_i : y \in P$.
- No noise: $\overline{y}$ optimal.
- Noise: perturbation of objective function.
- Design code, relaxation accordingly.

Tanner Graph

- The *Tanner Graph* of a linear code is a bipartite graph modeling the *parity check matrix* of the code.



- "Variable nodes" $y_1, \ldots, y_n$.
- "Check Nodes" $c_1, \ldots, c_m$.
- $N(j)$: n'hood of check c_j.
- Code words: $y \in \{0,1\}^n$ s.t.:

$$\forall c_j, \sum_{i \in N(j)} y_i = 0 \pmod 2$$

- Codewords: 0000000, 1110000, 1011001, etc.

IP/LP Formulation of ML Decoding

- Variables $\{f_i\}$ for each code bit y_i.

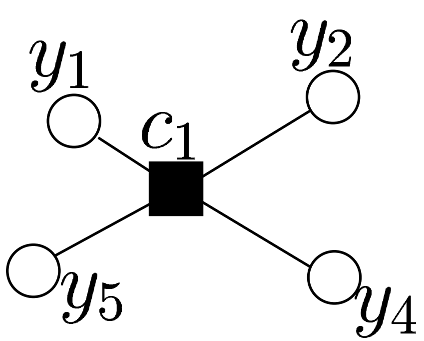
$$\text{IP: } f_i \in \{0, 1\}. \quad \text{LP: } 0 \leq f_i \leq 1.$$

- For check bit c_j, E_j = **valid configurations** of $N(j)$.

$$E_j = \{S \subseteq N(j) : |S| \text{ even}\}$$

- Variables $\{w_{j,S}\}$ for each check node c_j, $S \in E_j$.

$$\text{IP: } w_{j,S} \in \{0, 1\}. \quad \text{LP: } 0 \leq w_{j,S} \leq 1.$$



- Vars: $w_{1,\emptyset}$, $w_{1,\{1,2,4,5\}}$, $w_{1,\{1,2\}}$, $w_{1,\{1,4\}}$, $w_{1,\{1,5\}}$, $w_{1,\{2,4\}}$, $w_{1,\{2,5\}}$, $w_{1,\{4,5\}}$

IP/LP Formulation of ML Decoding

- Minimize $\sum_i \gamma_i f_i$, subject to:

$$\forall \text{ checks } j, \qquad \sum_{S \in E_j} w_{j,S} = 1$$

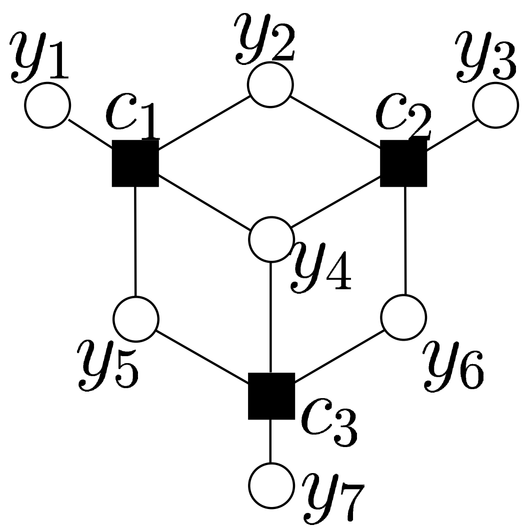
$$\forall \text{ edges } (i,j) \in G, \qquad f_i = \sum_{\substack{S \in E_j \\ S \ni i}} w_{j,S}$$

- Let P be the relaxed polytope.

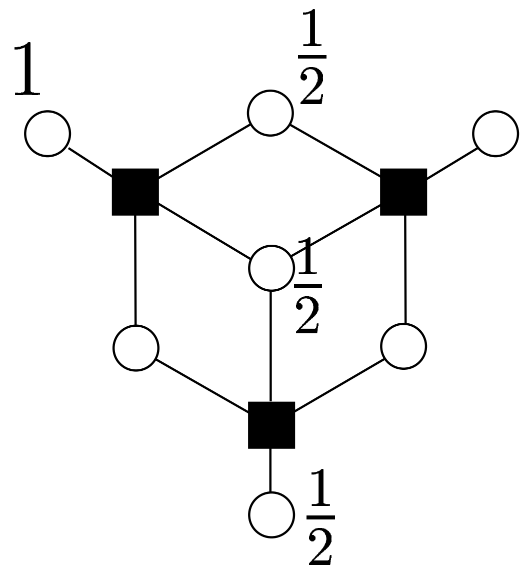
$$\text{code } C = \{ \ f \in \{0,1\}^n \mid \exists \ \ w \ \text{ s.t. } \ (f,w) \in P \ \}$$

- IP: formulation of ML decoding.
- What do fractional solutions look like?

Fractional Solutions



- Suppose: $$\begin{aligned}\gamma_1 &= -2.8\\ \gamma_2 &= +0.8\\ \gamma_{3..7} &= +1\end{aligned}$$
- ML codeword: $[1, 1, 1, 0, 0, 0, 0]$
- ML codeword cost: -1.



- Frac. sol: $f = [1, \frac{1}{2}, 0, \frac{1}{2}, 0, 0, \frac{1}{2}]$.
- $$\begin{aligned} w_{1,\{1,2\}} &= w_{1,\{1,4\}} = \tfrac{1}{2}\\ w_{2,\{2,4\}} &= w_{2,\emptyset} = \tfrac{1}{2}\\ w_{3,\{4,7\}} &= w_{3,\emptyset} = \tfrac{1}{2}\end{aligned}$$
- Frac. sol cost: -1.4.

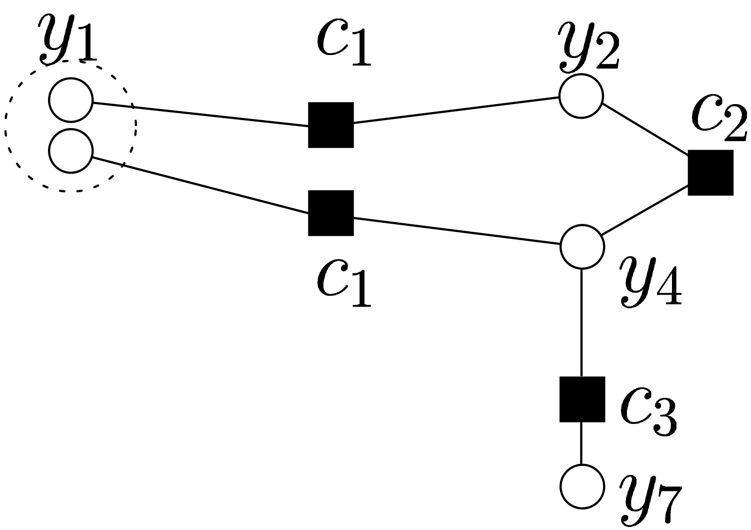
LP Decoding Success Conditions

- Pr[Decoding Success] = Pr[$\overline{y}$ is the unique OPT].
- Assume $\overline{y} = 0^n$
  - Common asssumption for linear codes.
  - OK in this case due to symmetry of polytope.
- Pr[$\overline{y}$ is the unique OPT]
  = Pr[All other solutionss have cost > 0].

Theorem [FKW, CISS '03]: Assume the all-zeros codeword was sent. Then, the LP decodes correctly iff all non-zero points in P have positive cost.

Pseudocodewords

- Pseudocodewords are scaled points in P.



- Previous example:
  $f = [1, \frac{1}{2}, 0, \frac{1}{2}, 0, 0, \frac{1}{2}]$.
- Scaled to integers:
  $f' = [2, 1, 0, 1, 0, 0, 1]$.

- Natural combinatorial definition of pseudocodeword (independent of LP relaxation).

Theorem [FKW, CISS '03]: LP decodes correctly iff all pseudocodewords have cost > 0.

Fractional Distance

- Classical distance:
  - δ = min Hamming dist. of codewords in C.
- Adversarial performance bound:
  - ML decoding can correct $\delta/2 - 1$ errors.
- Another way to define minimum distance:
  - δ_f = min (l_1) dist. between two integral verts of P.
- Fractional distance:
  - δ_f = min (l_1) dist. between an integral and a fractional vertex of P.
  - δ_f = min wt. fractional vertex of P.
  - Lower bound on classical distance: $\delta_f \leq \delta$.
  - LP Decoding can correct $\delta_f/2 - 1$ errors.

LP Decoding corrects $\delta_f/2 - 1$ errors

- Suppose fewer than $\delta_f/2$ errors occur.
- Let (f^*, w^*) be a vertex of P, $f^* \neq 0^n = \overline{y}$.
  $\sum_i f_i \geq \delta_f$.
- When $\overline{y} = 0^n$, $\gamma_i = -1$ if i flipped, +1 o.w.; So,

$$\sum_i \gamma_i f_i^* = \sum_{i \text{ not flipped}} f_i^* - \sum_{i \text{ flipped}} f_i^*$$

- Since $\displaystyle\sum_{i \text{ flipped}} f_i^* \;<\; \delta_f/2 \implies \sum_{i \text{ not flipped}} f_i^* > \delta_f/2.$

- Therefore $\sum_i \gamma_i f_i^* > 0.$

Computing the Fractional Distance

- Computing δ for linear/LDPC codes is NP-hard.
- If the polytope has small size (LDPC), the fractional distance is easily computed.
  - More general problem: Given an LP, find the *two* best vertices v, v'.
  - Algorithm:
    * Find v.
    * Guess the facet on which v' sits but v does not.
    * Set facet to equality, obtaining P'.
    * Minimize $g()$ over P'.
- Good approximation to the classical distance?
- Good prediction of relative classical distance?

Using Girth for Error Bounds

- For rate-1/2 RA (cycle) codes: If G has large girth, neg-cost pseudocodewords (promenades) are rare.
- Erdös (or [BMMS '02]): Hamiltonian 3-regular graph with girth $\log n$.

Theorem [FK, FOCS '02]: For any $\alpha > 0$, as long as $p < 2^{-4(\alpha+(\log 24)/2)}$, WER $\leq n^{-\alpha}$.

- Arbitrary G, girth g, all var. nodes have degree $\geq d$:

Theorem [FKW, CISS '03]: $\delta_f \geq (d-1)^{\lceil g/4 \rceil - 1}$

- Can achieve $\delta_f = \Omega(n^{1-\epsilon})$. Stronger graph properties (expansion?) are needed for stronger results.

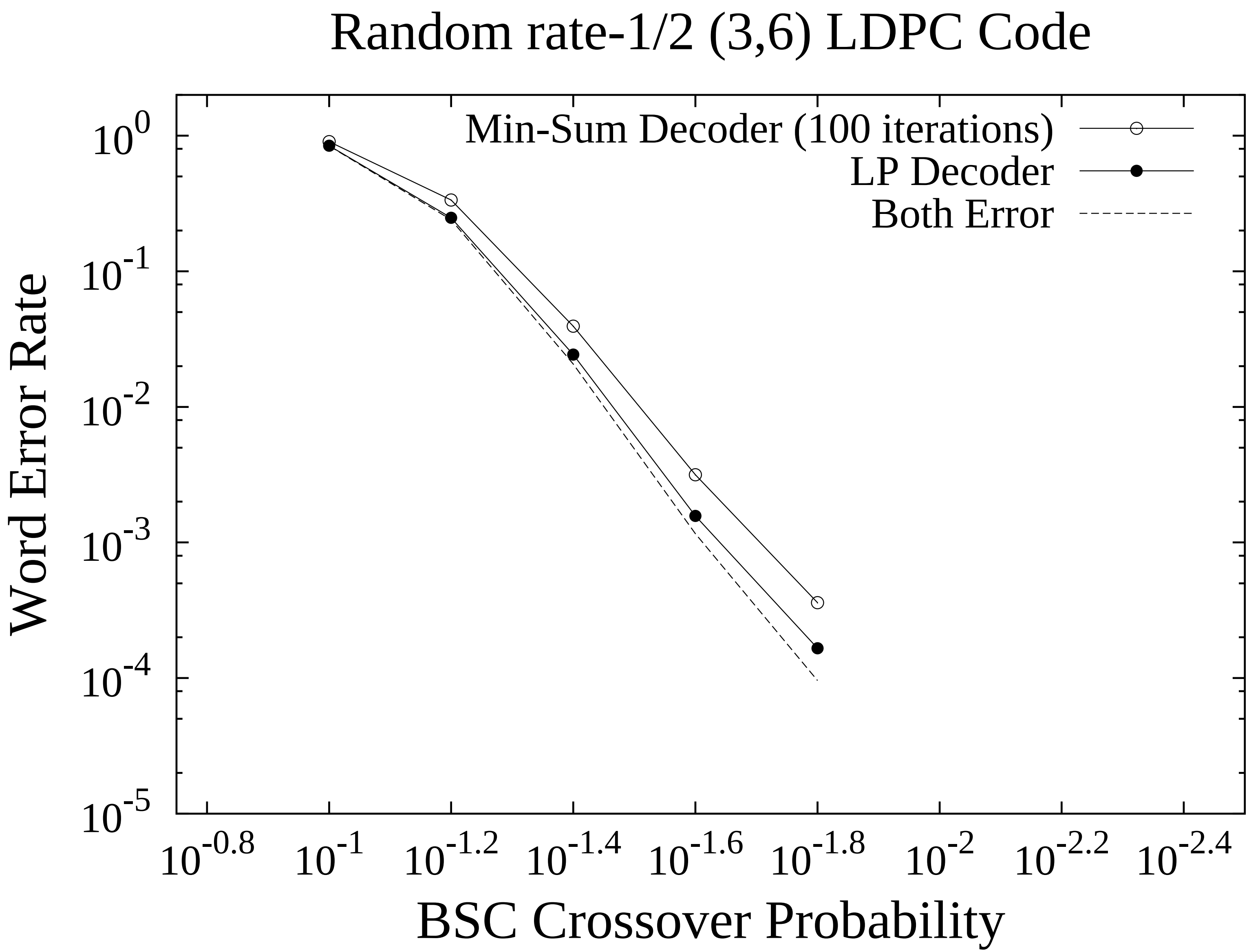
Other “pseudocodewords”

- BEC: Iterative decoding successful iff no zero-cost “stopping sets.” [DPRTU ’02]
  - In the BEC, pseudocodewords = stopping sets.
  - Iterative/LP decoding: same performance in BEC.
- Tail-Biting trellisses (TBT): Iterative decoding successful iff “dominant pseudocodeword” has negative cost [FKMT ’98].
  - TBT: need LP along lines of [FK, FOCS ’02].
  - Iterative/LP decoding: same performance on TBT.
- “Min-sum” decoding successful iff no neg-cost “deviation sets” in the computation tree [Wib ’96].
  - Pseudocodewords are natural “closed” analog of deviation sets.

Other Results

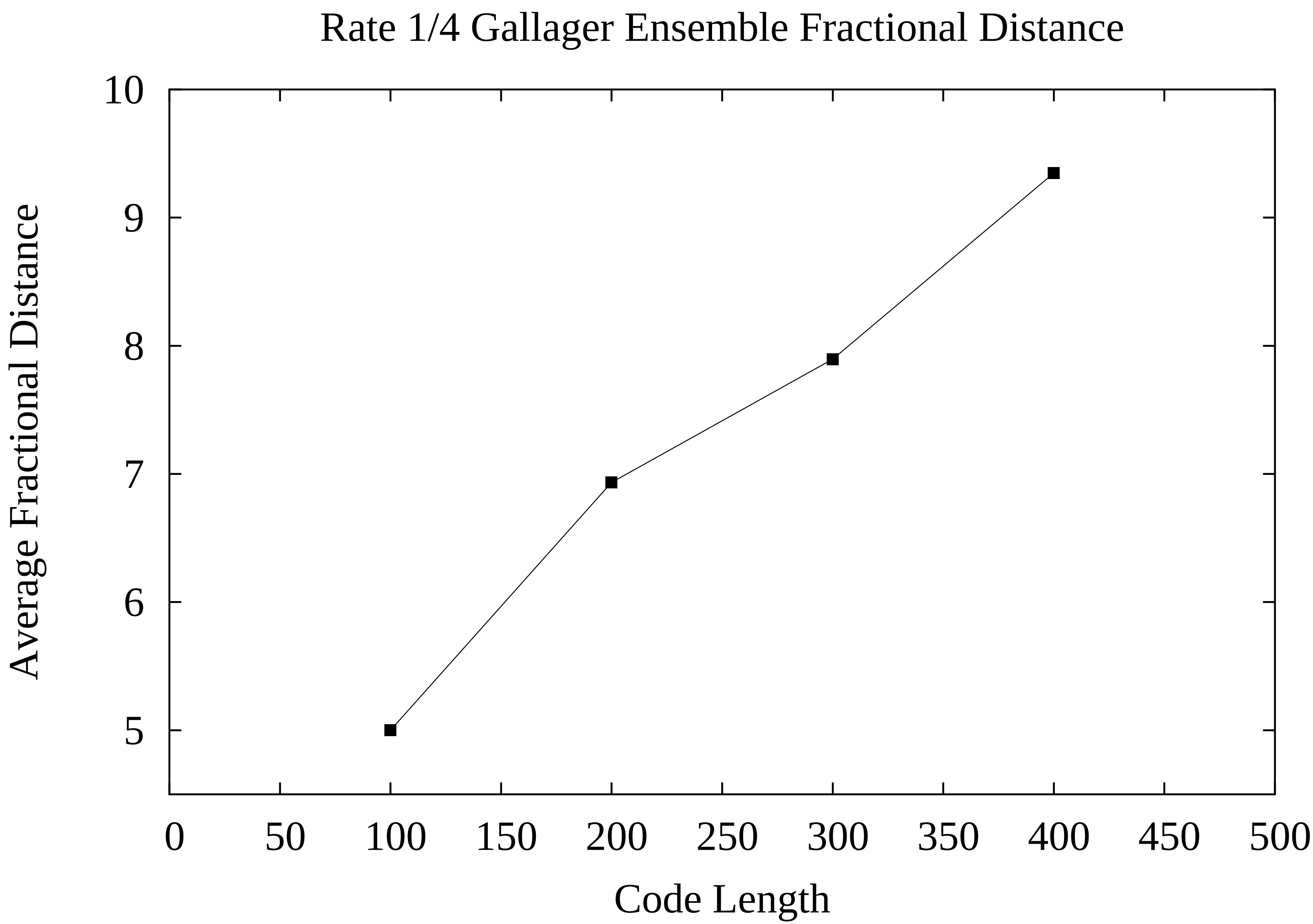
- For "high-density" binary linear codes, need representation of P without exponential dependence on check node degree.
  - Use "parity polytope" of Yannakakis ['91].
  - Orig. representation: $O(n + m2^{d_c})$.
  - Using parity polytopes: $O(mn + md_c^2 + nd_v d_c)$.
- New iterative methods [FKW, Allerton '02]:
  - Iterative "tree-reweighted max-product" [WJW '02] tries to solve dual of our LP.
  - Subgradient method for solving LP gives provably convergent iterative algorithm.
- Experiments on performance, distance bounds.

Performance Comparison



- Length 200, left degree 3, right degree 6.

Growth of Average Fractional Distance



- "Gallager" distribution, left degree 3, right degree 4.

Future Work

- New WER, fractional distance bounds:
  - Lower rate turbo codes (rate-1/3 RA).
  - Other LDPC codes, including
    - Expander codes, irregular LDPC codes, other constructible families.
  - Random LDPC, linear codes?
- ML Decoding using IP, branch-and-bound?
- Using generic "lifting" procedures to tighten relaxation?
- Deeper connections to "sum-product" belief-propagation?
- LP decoding of other code families, channel models?