FINITE LENGTH ANALYSIS OF LDPC CODES WITH LARGE LEFT DEGREES

Junan Zhang and Alon Orlitsky
Dept. of ECE, UCSD

Outline

- Existing Work
- New Algorithms
- Calculation Results
- Theoretical Results
- Conclusions

Background

Final Goal: Design LDPC codes with short length, low complexity and low error probability.

- *LDPC Code*: easy to decode, approach capacity of erasure channel with asymptotic large length.
- *Stopping Set*: A subset of variable nodes which has no check node connected with the subset just once.
- *Combinatorial Problem*: Stopping sets determine block and bit error probability over erasure channel.
- *Existing Work*: Di, Poietti, Telatar, Richardson and Urbanke gave exact expressions for block and bit error probability. Both need $O(n^5)$ computational complexity. They also gave efficient algorithms for left degree 2 and 3. The algorithm for left degree 2 is same as in our algorithm.
- *New algorithm*: Efficient algorithms for general left degrees. For block error probability, computational complexity is $O(n^3)$. For bit error probability, it is $O(n^4)$.

Results

- Efficient algorithms for *average* block and bit error probability applied to regular LDPC code ensembles.

- Extension to simple left-irregular LDPC code ensembles.

- Calculation results of block error probability for several LDPC codes.

- Theoretical block error behavior for small erasure probability
  - When block length n increases, average block error probability decreases as $\Theta(n^{-(\lceil\frac{l}{2}\rceil-1)})$. l is the lowest left degree of the code.
  - Proof for regular codes.

Error Probability Calculation (Di et al.)

Define:

- e: # erased variable nodes.
- m: size of the maximal stopping set contained in erased variable nodes.
- $T(e,c,0)$: # bipartite graphs connecting el edges from e erased variable nodes to c check nodes (may not used).
- $O(e,m,c,0)$: # bipartite graphs containing the maximal stopping set of size m.
- $N(e,c,0)$: # bipartite graphs containing no stopping set.

Average bit error probability of LDPC code ensemble:

$$E_{\mathcal{C}(n,x^{l-1},x^{r-1})}[P_b^{IT}(G,\epsilon)] = \sum_{e=1}^{n} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e} \sum_{m=1}^{e} \frac{m}{n} \cdot \frac{\binom{e}{m} O(e,m,c,0)}{T(e,c,0)}$$

Average block error probability of LDPC code ensemble:

$$E_{\mathcal{C}(n,x^{l-1},x^{r-1})}[P_B^{IT}(G,\epsilon)] = \sum_{e=0}^{c-1} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e} \cdot \left(1 - \frac{N(e,c,0)}{T(e,c,0)}\right) + \sum_{e=c}^{n} \binom{n}{e} \epsilon^e (\bar{\epsilon})^{n-e}$$

Error Probability Calculation (contd.)

- $O(e, m, c, 0)$ and $N(e, c, 0)$ can be calculated recursively:

$$
\begin{aligned}
T(e,c,d) &:= \binom{d+cr}{el}(el)! \\
N(e,c,d) &:= T(m,c,d) - \sum_{m=1}^{e} \binom{e}{m} O(e,m,c,d) \\
O(e,m,c,d) &:= \sum_{k=1}^{\min(c,\lfloor \frac{m \cdot l}{2} \rfloor)} \binom{c}{k} \mathtt{coef}(((1+x)^r - 1 - rx)^k (1+x)^d, x^{ml})(ml)! \cdot \\
&\quad N(e-m, c-k, d+kr-ml)
\end{aligned}
$$

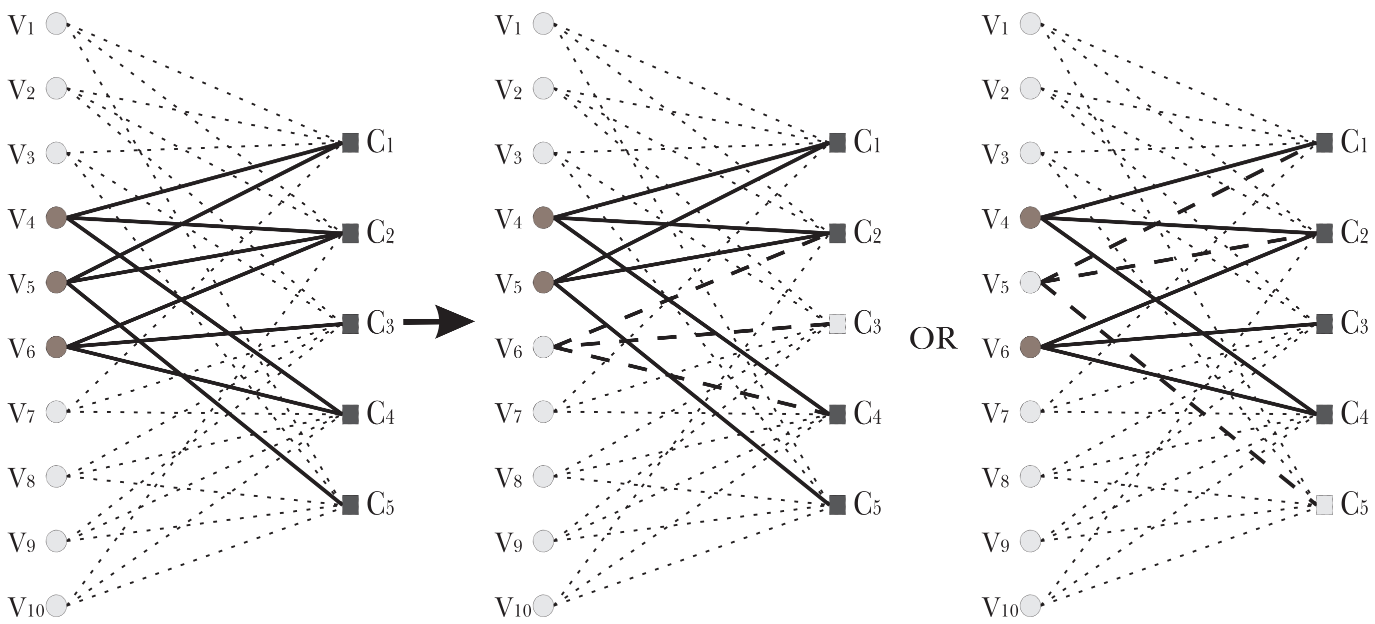
  with the boundary condition $O(e, m, c, d) = 0$ if $e \leq 0$ or $ml > cr + d$.

- Holds for regular codes.

- $O(n^5)$ computational complexity, feasible for small block length.

Efficient Algorithm for $l = 2, 3$ (Di et al.)

- *Motivation*
  The method is recursion on bipartite graphs containing no stopping set. Any bipartite graph containing no stopping set must have at least one check node with degree one. Delete one such check node and its connected variable node, also remove any edges connected with these two nodes. The new graph will still have no stopping set.



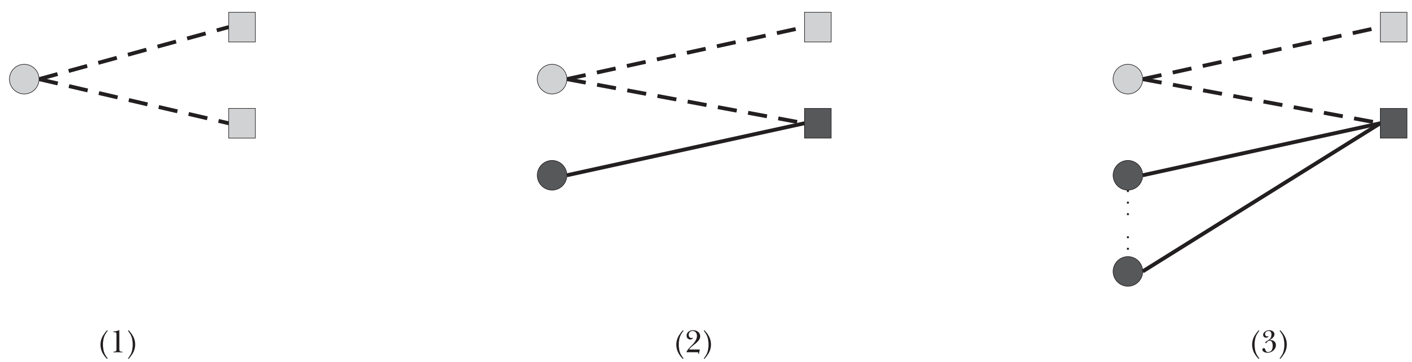
Efficient Algorithm for $l = 2, 3$ (contd.)

- *Calculation*
  Efficient recursion for $N(e, c, d)$ when $l = 2$ and $l = 3$. For example, when $l = 2$,

$$N(v, c, d) = (l!)^v v! \sum_{u=1}^{c} \sum_{s=1}^{u} a_2(v, u, s, d) \frac{c!}{(c-v)!} r^u (r-1)^{u-s}$$

where $a_2(v, u, s, d)$,for $v \geq 1, u \geq 0, s \geq 1$, can be calculated by recursion:

$$\begin{aligned} a_2(v, u, s, d) \cdot s \quad &= \quad a_2(v-1, u-2, s-2, d) + \\ &\quad\ a_2(v-1, u-1, s, d) \cdot s + \\ &\quad\ a_2(v-1, u-1, s-1, d)[(u-1)r + d - (s-1)(r-1) - (v-1)l] \end{aligned}$$



with the boundary condition $a_2(v = 0, u, s, d) = \begin{cases} 1, & u = 0, s = 0, d \geq 0, \\ 0, & \text{otherwise.} \end{cases}$

New Algorithm

Theorem 1 *For $l \geq 2$*

$$N(v,c,d) = (l!)^v v! \sum_{u=1}^{c} \sum_{s=1}^{u} a_l(v,u,s,d) \frac{c!}{(c-u)!} r^u (r-1)^{u-s}.$$

Notation:

- $N(v,c,d)$: # non-stopping-set bipartite graphs connecting el edges from e erased variable nodes to c check nodes (may not used) and the super check node with degree d (only for bit error probability).

- $a_l(v,u,s,d)$: # non-stopping-set bipartite graphs having u check nodes used, among which there are s nodes with degree one. But permutation of variable nodes, check nodes and sockets in nodes has been ignored.

New Algorithm (contd.)

Calculate $a_l(v, u, s, d)$ recursively:

$$a_l(v+1, u', s', d) \cdot s' = \sum_{q_1^I, s_1^J, t} a_l(v, u, s, d) \cdot F(q_1^I, s_1^J, t, e_d)$$

where $u' = u + \sum_{i=1}^{I} q_i + 1, \quad s' = s - \sum_{j=1}^{J} s_j + (q_1 + 1), \quad e_d = ur + d - s(r-1) - vl.$

and $I \stackrel{\text{def}}{=} \min(l-1, r),\ J \stackrel{\text{def}}{=} \min(l-1, r-1).$

with boundary condition $a_l(0, u, s, d) = \begin{cases} 1 & u = 0, s = 0, d \geq 0 \\ 0 & \text{otherwise,} \end{cases}$

Notation:

- $[q_1^I, s_1^J, t]$ describe all the patterns to add a new variable node and new check node(s).

- $q_2, \ldots, q_I$: # new check nodes of degree-i.

- q_1: one less than # new degree-1 check nodes (ensure at least one exists).

- $s_1, \ldots, s_J$: # old degree-1 check nodes to which we add i new edges.

- t: # edges we add to the old check nodes with degree ≥ 2 and the super check node.

New Algorithm (contd.)

Patterns $[q_1^I, s_1^J, t]$ *and Factor* $F(q_1^I, s_1^J, t, e_d)$*:*

- Pattern $[q_1^I, s_1^J, t]$ satisfies

$$\begin{aligned} \sum_{i=1}^{I} i \cdot q_i + \sum_{j=1}^{J} j \cdot s_j + t &= l - 1 \\ \sum_{i=1}^{J} s_i &\leq s \\ t &\leq e_d \end{aligned}$$

- $$F(q_1^I, s_1^J, t, e_d) = (q_1 + 1) \cdot \frac{\prod_{i=1}^{I} [Q(i)]^{q_i}}{(q_1+1)! \prod_{i=2}^{I} q_i!} \cdot \prod_{m=1}^{\sum_{j=1}^{J} s_j} (s - m + 1) \cdot \frac{\prod_{j=1}^{J} [S(j)]^{s_j}}{\prod_{j=1}^{J} s_j!} \cdot \binom{e_d}{t}$$

  where

$$Q(i) = \begin{cases} \frac{(r-2)\cdots(r-(i-1))}{i!} & i \geq 3 \\ \frac{1}{2} & i = 2 \\ 1 & i = 1 \end{cases}$$

$$S(j) = \begin{cases} \frac{(r-2)\cdots(r-j)}{j!} & j \geq 2 \\ 1 & j = 1 \end{cases}$$

New Algorithm (contd.)

Complexity and Extension to Simple Left-irregular Codes:

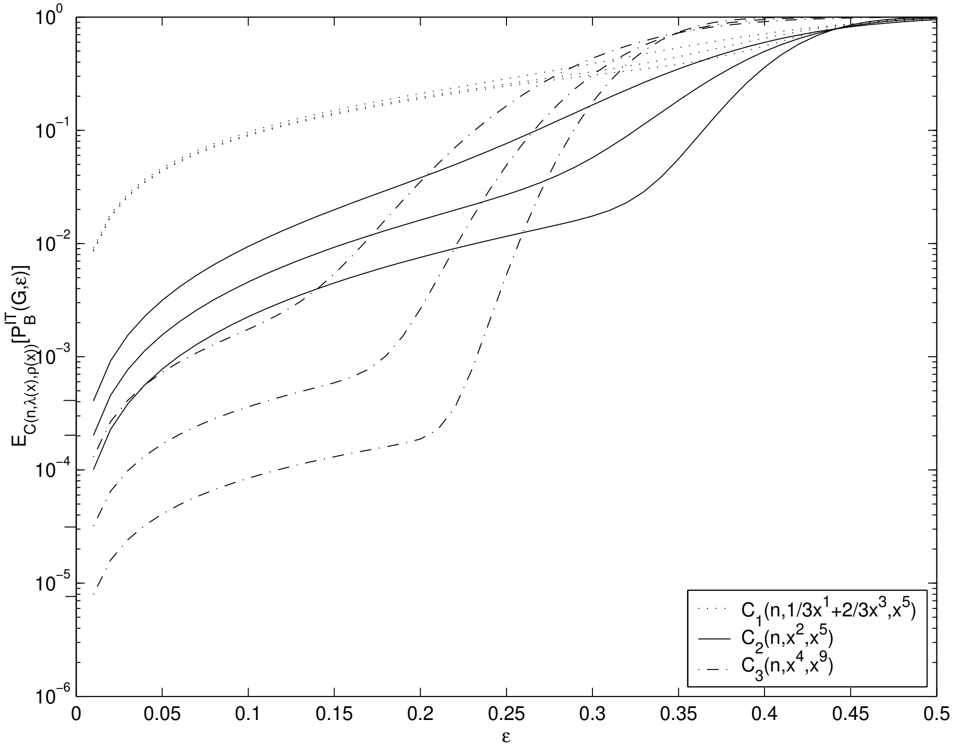
- For block error probability, need only $d = 0$. Computational complexity $O(n^3)$. Can calculate for block length. e.g. 5000.

- For bit error probability, need $d > 0$. Computational complexity $O(n^4)$.

- Recursion holds for all regular LDPC codes and can be extended to left-irregular LDPC codes after introducing $[v_{l_1}, v_{l_2}, \cdots, v_{l_d}]$ to describe variable nodes with different degrees. But requires higher computational complexity.

- Several techniques to accelerate the computation, including decomposing $[q_1^I, s_1^J, t]$ into three parts and doing the calculation in log-domain.

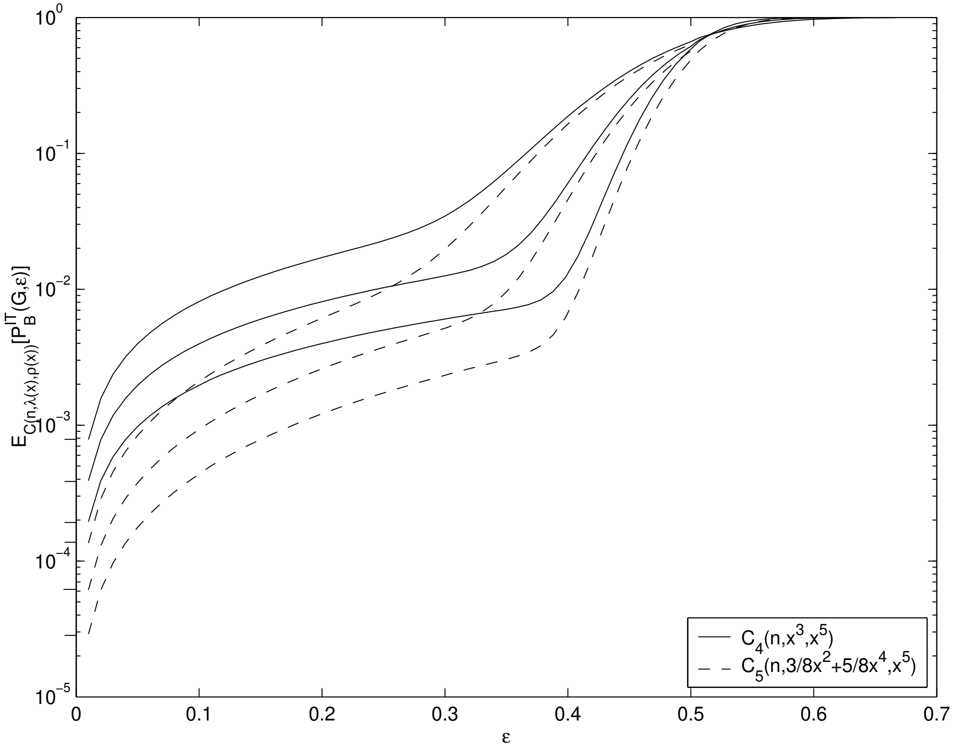
Results: Average Block Error Probability for Several LDPC Codes



$E_{\mathcal{C}_{1,2,3}(n)}[P_B^{IT}(G,\epsilon)]$ as a function of ϵ.

For each code, upper, middle and lower curves correspond to $n = 64$, $n = 128$ and $n = 256$.

Results: Average Block Error Probability for Several LDPC Codes (contd.)



$E_{\mathcal{C}_{4,5}(n)}[P_B^{IT}(G,\epsilon)]$ as a function of ϵ.

For each code, upper, middle and lower curves correspond to $n = 60$, $n = 120$ and $n = 240$.

Theoretical Block Error Behavior

Theorem 2 For regular LDPC code $\mathcal{C}(n, x^{l-1}, x^{r-1})$, $\exists\ \epsilon_0(l,r) > 0$, $\forall\ \epsilon < \epsilon_0(l,r)$

$$\lim_{n\to+\infty} E_{\mathcal{C}(n,\lambda(x),\rho(x))}[P_B^{IT}(G,\epsilon)] = \begin{cases} \epsilon\Theta(n^0) & l = 2, \\ \epsilon\Theta(n^{-1}) & l = 3,4, \\ \epsilon\Theta(n^{-(\lceil\frac{l}{2}\rceil-1)}) & l \geq 5 \end{cases}$$

Proof:

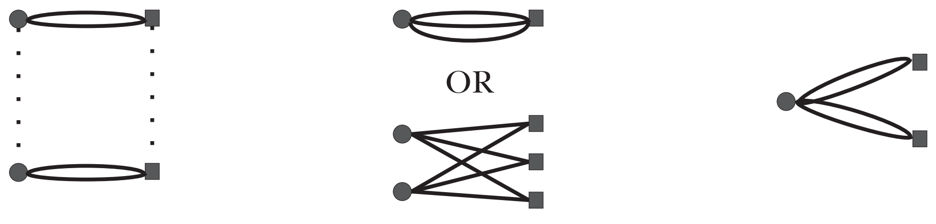
- Upper Bound

$$\begin{aligned} E_{\mathcal{C}(n,x^{l-1},x^{r-1})}[P_B^{IT}(G,\epsilon)] &\leq \sum_{S\subseteq V,\ 1\leq v\overset{\text{def}}{=}|S|\leq 2\epsilon n} \epsilon^v \frac{\text{coef}(((1+x)^r - rx)^c, x^{vl})}{\binom{n\cdot l}{v\cdot l}} + Pr(e > 2\epsilon n) \\ &\leq \sum_{v=1}^{2\epsilon n} \epsilon^v \binom{n}{v} \frac{\binom{c}{\lfloor\frac{vl}{2}\rfloor}}{\binom{nl}{vl}} r^{vl} + e^{-n\epsilon/3} \end{aligned}$$

1. For small fixed v, $\binom{n}{v}\frac{\binom{c}{\lfloor\frac{vl}{2}\rfloor}}{\binom{nl}{vl}}$ behaves as $\Theta(n^{-(\lceil\frac{vl}{2}\rceil-v)}) = \begin{cases} \Theta(n^0) & l = 2 \\ \Theta(n^{-1}) & \begin{array}{c} l = 3,\ v = 1,2 \\ \text{or} \\ l = 4,\ v = 1 \end{array} \end{cases}$.

Proof (contd.)

Which correspond to



$l = 2$ $l = 3$ $l = 4$

Bad Small Connections

2. Items $\epsilon^v \binom{n}{v} \frac{\binom{c}{\lfloor \frac{vl}{2} \rfloor}}{\binom{nl}{vl}} r^{vl}$ from $v = 1, \cdots, 2\epsilon n$ form a decreasing geometric series for $l > 2$ and ϵ_0 small enough.

Therefore, $E_{\mathcal{C}(n,x^{l-1},x^{r-1})}[P_B^{IT}(G,\epsilon)] \leq \epsilon\Theta(n^{-(\lceil \frac{l}{2} \rceil - 1)}) + e^{-n\epsilon/3}$ for ϵ small enough.

Proof (contd.)

- Lower Bound:

$$E_{\mathcal{C}(n,x^{l-1},x^{r-1})}[P_B^{IT}(G,\epsilon)] \quad > \quad Pr(\bigcup_{v\in V} \text{node } v \text{ is a size-1 stopping set})$$

by using $\Pr(\bigcup_{i=1}^n A_i) \geq \sum_{i=1}^n Pr(A_i) - \sum_{1\leq i<j\leq n} Pr(A_i \bigcap A_j)$

$$\begin{aligned} & Pr(\bigcup_{v\in V} \text{node } v \text{ is a size-1 stopping set}) \\ \geq \quad & \epsilon\Theta(n^{-(\lceil \frac{l}{2}\rceil -1)}) - \binom{n}{2}\epsilon^2 \frac{\text{coef}(((1+x+y)^r - rx(1+y)^{r-1} - ry(1+x)^{r-1} + r(r-1)xy)^c, x^l y^l)}{\binom{n\cdot l}{2l}} \\ \geq \quad & \epsilon\Theta(n^{-(\lceil \frac{l}{2}\rceil -1)}) - \binom{n}{2}\epsilon^2\Theta(n^{-2\lceil \frac{l}{2}\rceil}) \qquad \square \end{aligned}$$

- Conjecture: also holds for irregular code by replacing l by the lowest left degree.

Conclusions and Future Work

- Efficient recursion: calculation of average block error probability for regular and simple left-irregular LDPC codes.
  - ⋆ Union bound method can give a very tight result for small ϵ. However it is loose around "waterfall" region[2].
  - ? Algorithm simpler than exact expression but tighter than union bound.

- Bad small connections dominate the average block error probability of LDPC code ensembles. It is bad, especially for irregular codes with degree-2 variable nodes.
  - ? Exact block error probability for other (expurgated) ensembles of LDPC codes which avoid bad small connection.
  - ? Given n variable nodes, c check nodes and E edges, find structures or strategies to construct good finite length LDPC codes.
    e.g. Will the code which maximizes the smallest stopping set be irregular?

Acknowledgment

We thank Rüdiger Urbanke for introducing us to finite-length analysis and for insightful discussions.

Reference

[1] C. Di and D. Proietti and T. Richardson and E. Telatar and R. Urbanke. Finite length analysis of low-density parity-check codes. Submitted to IEEE Transactions on Information Theory.

[2] T. Richardson, A. Shokrollahi and R. Urbanke. Finite length analysis of various low-density-parity-check ensembles for the binary erasure channel. Preprint.