Sum-Product decoding algorithms using linear forms

Bertrand Ndzana Ndzana

Joint work with
Harm Cronie and Amin Shokrollahi

EPFL, Switzerland

Algo Workshop, 2009

Introduction

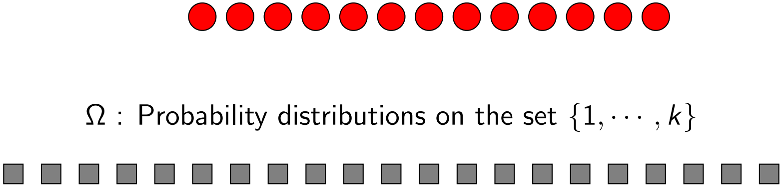
- Binary sparse graph codes can be decoded by belief propagation (BP).
- Extension of BP to GF(q) is straightforward [1]
  - However, computationally a problem for large q.
- **How can we decode efficiently on non-binary channels using binary codes ?**

[1] *Davey and MacKay, 1998*

Prerequisites: LT codes (Luby, 1998)

- First class of Fountain Codes
- Parameters (k, Ω)
- Decoding complexity : $O(k \log(k))$

k input symbols

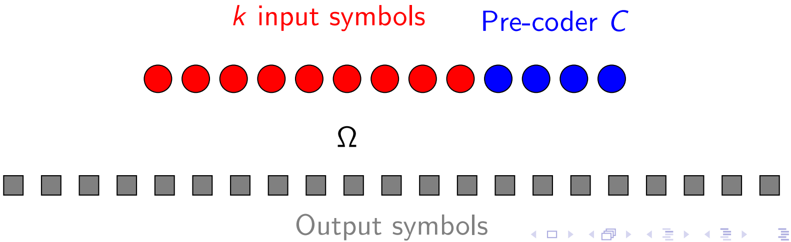


Ω : Probability distributions on the set $\{1, \cdots, k\}$

Output symbols

Prerequisites: Raptor codes (Shokrollahi, 2003)

- Parameters (k, C, Ω)
- Achieve constant average degree and vanishing probability of error
- Decoding : Belief-Propagation (Etesami and Shokrollahi, 2005)
- Decoding complexity : $O(k)$



Prerequisites: field $\mathrm{GF}(q)$

- Field $\mathrm{GF}(q)$: $q = 2^m$
  - $\mathrm{GF}(q) = \{0, \cdots, q-1\}$
  - $$\begin{cases} \mathrm{GF}(q) & \rightarrow \mathrm{GF}(2)^m \\ x & \mapsto b(x) = [b_1(x), \cdots, b_m(x)] \end{cases}$$

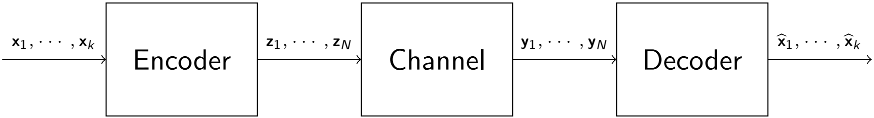
Prerequisites: linear forms

- Linear form $\varphi : \mathrm{GF}(2)^m \to \mathrm{GF}(2)$,

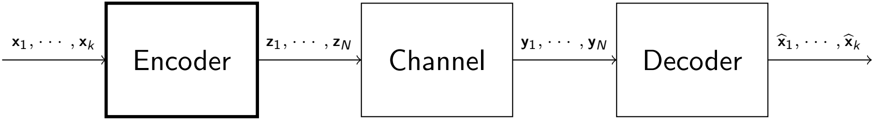
$$\varphi(x) = \varphi(b(x)) = \varphi^{(1)} b_1(x) + \cdots + \varphi^{(m)} b_m(x)$$

where $\varphi^{(i)} \in \mathrm{GF}(2)$

q-ary communication system

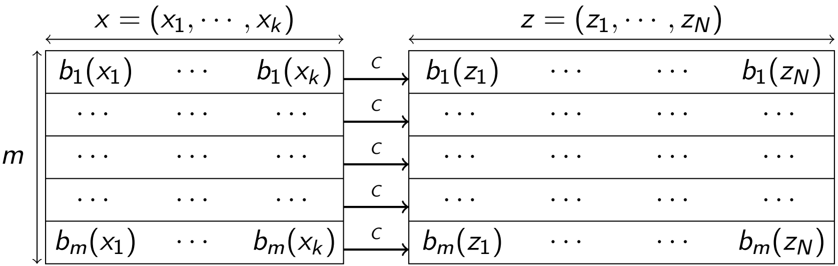


Encoding

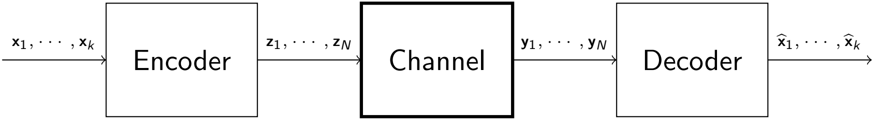

$\mathbf{x}_1, \cdots, \mathbf{x}_k$
Encoder
$\mathbf{z}_1, \cdots, \mathbf{z}_N$
Channel
$\mathbf{y}_1, \cdots, \mathbf{y}_N$
Decoder
$\widehat{\mathbf{x}}_1, \cdots, \widehat{\mathbf{x}}_k$

Encoding

- C: $[N, k]_2$-code, $C_q = C \otimes \mathrm{GF}(q)$.

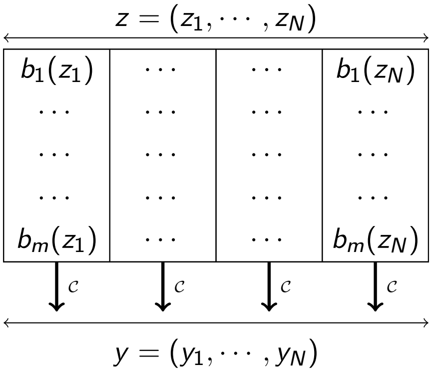


Transmission

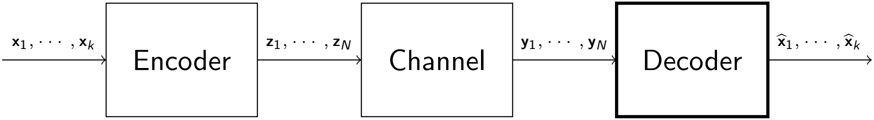


Transmission

- $\mathcal{C}$ q-ary channel.

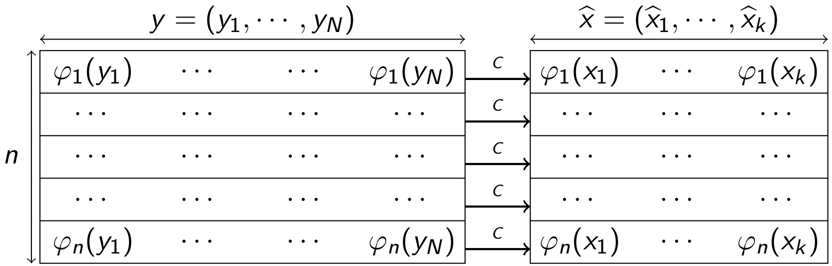


Decoding


$\mathbf{x}_1, \cdots, \mathbf{x}_k$
Encoder
$\mathbf{z}_1, \cdots, \mathbf{z}_N$
Channel
$\mathbf{y}_1, \cdots, \mathbf{y}_N$
Decoder
$\widehat{\mathbf{x}}_1, \cdots, \widehat{\mathbf{x}}_k$

Decoding

- Set of n linear forms $\{\varphi_1, \ldots, \varphi_n\}$, where $n \geq m$



Raptor codes

- Let a Raptor code with parameters (k, C, Ω) be given, where C is a precode of dimension k.
- We view this code as a code over $\mathrm{GF}(q)$.
- Formally, the Raptor code is tensored as a $\mathrm{GF}(2)$ module with the $\mathrm{GF}(2)$ module $\mathrm{GF}(q)$.
- Input symbols are from $\mathrm{GF}(q)$ and output symbols are binary linear combinations of the inputs symbols.

Raptor codes

Key observation

- Let $x_1, \ldots, x_k$ be k source symbols of a Raptor code with parameters (k, C, Ω).
- Let $y_1, \ldots, y_N$ be N output symbols received.
- Let G denote the corresponding decoding graph.
- Then for every linear form $\varphi \colon \mathrm{GF}(q) \to \mathrm{GF}(2)$ the graph G is the decoding graph between the input symbols $\varphi(x_1), \ldots, \varphi(x_k)$ and the output symbols $\varphi(y_1), \ldots, \varphi(y_N)$.

Raptor codes

- Let $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ be a set of n linear forms.
- Assume that $\dim\langle\Phi\rangle = m$.
- The code C_Φ is defined as

$$C_\Phi = \{(\varphi_1(x), \ldots, \varphi_n(x)) \mid x \in \mathrm{GF}(q)\}.$$

This is a linear code of blocklength n and dimension m.

Raptor codes

- A $m \times n$ generator matrix $G(C_\Phi)$ for the $[n, m]_2$-code C_Φ is given as

$$G(C_\Phi) = \begin{bmatrix} \varphi_1^{(1)} & \dots & \varphi_n^{(1)} \\ \vdots & \vdots & \vdots \\ \varphi_1^{(m)} & \dots & \varphi_n^{(m)} \end{bmatrix}$$

Raptor codes

Examples

- $\mathrm{GF}(q) = \mathrm{GF}(2^2)$.
  Let $\Phi = \{\varphi_1, \varphi_2, \varphi_3\}$ where

$$G(C_\Phi) = \begin{bmatrix} \varphi_1^{(1)} & \varphi_2^{(1)} & \varphi_3^{(1)} \\ \varphi_1^{(2)} & \varphi_2^{(2)} & \varphi_3^{(2)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

  C_Φ is a $[3,2]_2$-code called Hadamard-code.

Raptor codes

Examples

- $\mathrm{GF}(q) = \mathrm{GF}(2^4)$.
  Let $\Phi = \{\varphi_1, \varphi_2, \varphi_3\, \varphi_4, \varphi_5\, \varphi_6, \varphi_7\}$ where

$$G(C_\Phi) = \begin{bmatrix} \varphi_1^{(1)} & \cdots & \varphi_7^{(1)} \\ \varphi_1^{(2)} & \vdots & \varphi_7^{(2)} \\ \varphi_1^{(3)} & \vdots & \varphi_7^{(3)} \\ \varphi_1^{(4)} & \cdots & \varphi_7^{(4)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

  C_Φ is a $[7, 4]_2$-code called Hamming-code.

Decoding $\mathrm{GF}(2^m)$ Raptor Codes

Decoding

- The output symbols $z_1, z_2, \ldots z_N$ are generated and transmitted on the q-ary channel.
- Let $P(z|y)$ denote the posterior probability of having sent z given y.
- **Marginalized Raptor** : $\Pr[\varphi(z) = 0|y] = \sum_{\substack{u \in \mathrm{GF}(q) \\ \varphi(u)=0}} P(u \mid y)$.

Decoding $\mathrm{GF}(2^m)$ Raptor Codes

Algorithm1

1. Initialization : Initialize the values of output symbol y with the vector $\left(\ln\left(\frac{\Pr[\varphi_1(z)=0]}{1-\Pr[\varphi_1(z)=0]}\right), \ldots, \ln\left(\frac{\Pr[\varphi_n(z)=0]}{1-\Pr[\varphi_n(z)=0]}\right)\right)$
2. BP : Perform several BP decoding iterations.
3. ML: Perform ML-decoding of C_Φ for every input symbol.

Decoding $\mathrm{GF}(2^m)$ Raptor Codes

Algorithm2

1. Initialization : Initialize the values of each output symbol as done in Algorithm1.
2. BP : Perform t (design parameter) BP decoding iterations.
3. ML : Update probabilities on input nodes using ML-decoding of C_Φ.
4. Go to BP (or stop the process at preset number of iterations).

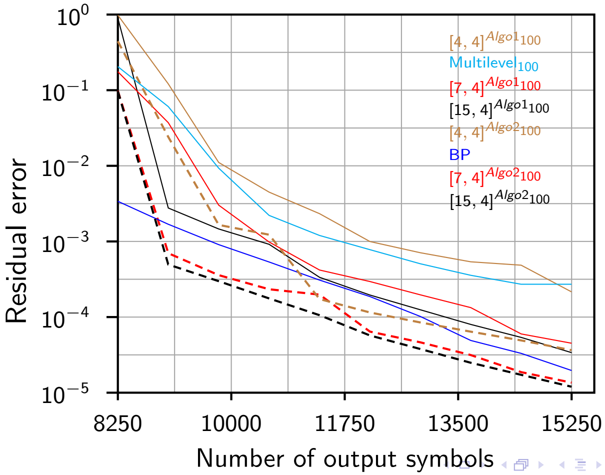
Decoding $\mathrm{GF}(2^m)$ Raptor Codes

Complexity

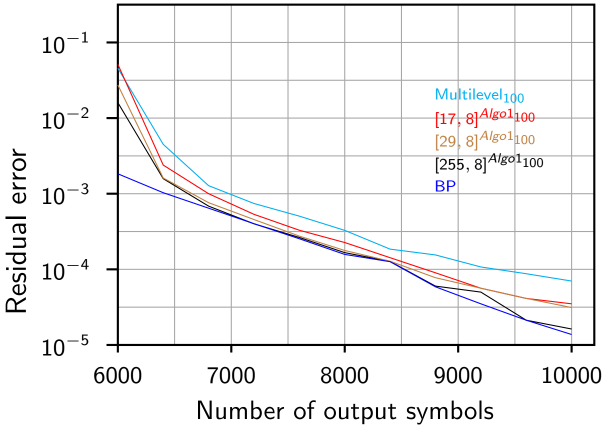
- **Our algorithm** : $O(Nn + m2^m)$
  - Typical case : $O(Nm)$
  - Worst case : $O(N2^m)$
- Belief-Propagation[2]: $O(Nm2^m)$

[2] *Davey and MacKay, 1998*

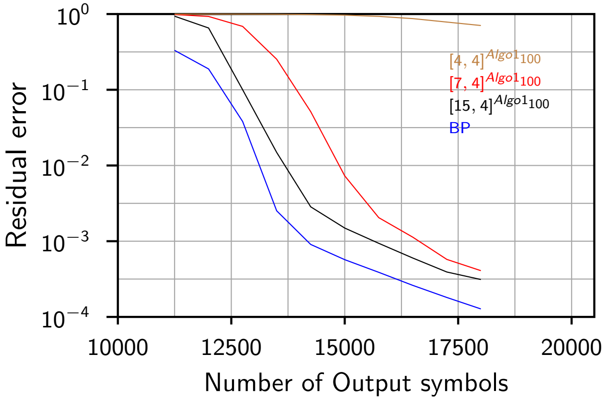
Simulation results(16-ary symmetric channels)



Simulation results(256-ary symmetric channels)



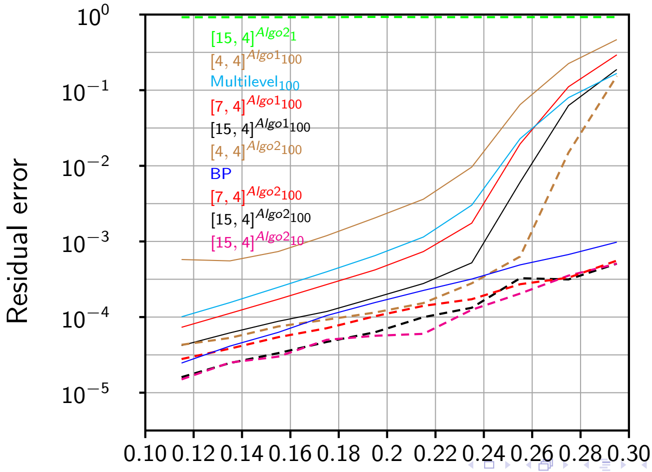
Simulation results(16-pam channels)



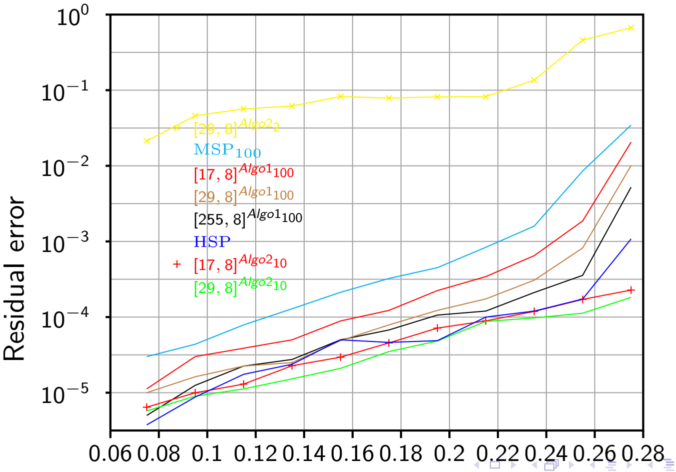
Conclusion

- Decoding binary Raptor codes on non-binary channels
  - Tradeoff between the running time and the decoding capability
- Applicable to any binary linear code
- Applicable to non-standard channels
- Can not be used when the underlying code is not binary

Simulation results(16-ary symmetric channels)



Simulation results(256-ary symmetric channels)



Simulation results (16-pam channels)