Performance Enhancements for Algebraic Soft Decision Decoding of Reed-Solomon Codes

ISIT 2004

Mostafa El-Khamy, Robert J. McEliece, and Jonathan Harel

{mostafa, rjm, harel}@systems.caltech.edu

California Institute of Technology

Pasadena, CA 91125, USA

Introduction

- Goal:

What is the ultimate performance of ASD of RS Codes?

Introduction

- Goal:

What is the ultimate performance of ASD of RS Codes?

$\boxed{\text{Find better Multiplicity Assignment algorithms}}$

Introduction

- Goal:

  What is the ultimate performance of ASD of RS Codes?

  $\boxed{\text{Find better Multiplicity Assignment algorithms}}$

- Existing Multiplicity Assignment Algorithms for ASD:

Introduction

- Goal:

  What is the ultimate performance of ASD of RS Codes?

  Find better Multiplicity Assignment algorithms

- Existing Multiplicity Assignment Algorithms for ASD:
  - Kötter and Vardy Algorithm [KV03]
  - Gaussian Approximation [PV03]

Outline

- Introduction
- Algebraic Soft Decoding
- Multiplicity Assignment Problem
- Soft Multiplicity Matrices
- The Chernoff Bound
- The Lagrangian
- Iterative Algorithm
- Numerical results & Conclusions

Introduction

- n-dimensional vector over F: $\mathbf{u} = (u_1, \ldots, u_n)$

Introduction

- n-dimensional vector over F: $\mathbf{u} = (u_1, \ldots, u_n)$
- $q \times n$ arrays: $W = (w_i(\beta))$, where $i = 1, \ldots, n$ and $\beta \in F$.

Introduction

- n-dimensional vector over F: $\mathbf{u} = (u_1, \ldots, u_n)$
- $q \times n$ arrays: $W = (w_i(\beta))$, where $i = 1, \ldots, n$ and $\beta \in F$.
- *Cost*:

$$|W| \triangleq \frac{1}{2} \sum_{i=1}^{n} \sum_{\beta \in F} w_i(\beta) \left(w_i(\beta) + 1 \right).$$

Introduction

- n-dimensional vector over F: $\mathbf{u} = (u_1, \ldots, u_n)$
- $q \times n$ arrays: $W = (w_i(\beta))$, where $i = 1, \ldots, n$ and $\beta \in F$.
- *Cost*:

$$|W| \triangleq \frac{1}{2} \sum_{i=1}^{n} \sum_{\beta \in F} w_i(\beta) \left(w_i(\beta) + 1 \right).$$

- *Score*:

$$\langle \mathbf{u}, W \rangle \triangleq \sum_{i=1}^{n} w_i(u_i).$$

Introduction

- n-dimensional vector over F: $\mathbf{u} = (u_1, \ldots, u_n)$
- $q \times n$ arrays: $W = (w_i(\beta))$, where $i = 1, \ldots, n$ and $\beta \in F$.
- *Cost*:

$$|W| \triangleq \frac{1}{2} \sum_{i=1}^{n} \sum_{\beta \in F} w_i(\beta) \left(w_i(\beta) + 1 \right).$$

- *Score*:

$$\langle \mathbf{u}, W \rangle \triangleq \sum_{i=1}^{n} w_i(u_i).$$

- $\mathbf{c} \in \mathbb{C}$ will be an (n, k, d) Reed-Solomon code over F.

Introduction

- n-dimensional vector over F: $\mathbf{u} = (u_1, \ldots, u_n)$
- $q \times n$ arrays: $W = (w_i(\beta))$, where $i = 1, \ldots, n$ and $\beta \in F$.
- *Cost*:

$$|W| \triangleq \frac{1}{2} \sum_{i=1}^{n} \sum_{\beta \in F} w_i(\beta) \left(w_i(\beta) + 1\right).$$

- *Score*:

$$\langle \mathbf{u}, W \rangle \triangleq \sum_{i=1}^{n} w_i(u_i).$$

- $\mathbf{c} \in \mathbb{C}$ will be an (n, k, d) Reed-Solomon code over F.
- *Channel Output*: APP matrix:

$$\Pi = (\pi_i(\beta)) = (Pr\left\{c_i = \beta | r_i\right\})$$

Algebraic Soft Decoding

$$\Pi \Rightarrow \boxed{\text{Multiplicity Assignment Algorithm; } A} \Rightarrow M$$

- $\mathbf{c} \rightarrow \Pi \xrightarrow{A} M$
- $m_i(\beta)$ is a non-negative integer.

Algebraic Soft Decoding

$\Pi \Rightarrow$ $\boxed{\text{Multiplicity Assignment Algorithm; } A}$ $\Rightarrow M$

- $\mathbf{c} \to \Pi \stackrel{A}{\to} M$
- $m_i(\beta)$ is a non-negative integer.
- $\mathbf{c}$ is on the GS list if

$$\langle \mathbf{c}, M \rangle > D_{k-1}(|M|) \equiv \mathbf{c} \vdash M$$

- $$D_v(\gamma) \leq -\frac{v}{2} + \sqrt{2v\gamma} + \frac{v^{3/2}}{8\sqrt{2\gamma}}.$$

Algebraic Soft Decoding

- $\Pr\{\mathcal{E}_A\} = \sum_{\Pi \in \overline{APP}} \Pr\{\mathcal{E}_A | \Pi\} \Pr\{\Pi\};\ \ \mathcal{E}_A = \{\mathbf{c} \nvdash M\}$

Algebraic Soft Decoding

- $\Pr\{\mathcal{E}_A\} = \sum_{\Pi \in \overline{APP}} \Pr\{\mathcal{E}_A | \Pi\} \Pr\{\Pi\}; \quad \mathcal{E}_A = \{\mathbf{c} \not\vdash M\}$
- Theorem:

$$\Pr\{\mathcal{E}_A | \Pi\} = \frac{1}{\sum_{\mathbf{c} \in \mathbb{C}} \mathbf{P}(\mathbf{c})} \sum_{\mathbf{c} \in \mathbb{C}} \Delta[\mathbf{c} \not\vdash M] \, \mathbf{P}(\mathbf{c}).$$

Algebraic Soft Decoding

- $\Pr\{\mathcal{E}_A\} = \sum_{\Pi \in \overline{APP}} \Pr\{\mathcal{E}_A | \Pi\} \Pr\{\Pi\}\,; \quad \mathcal{E}_A = \{\mathbf{c} \not\vdash M\}$
- Theorem:

$$\Pr\{\mathcal{E}_A | \Pi\} = \frac{1}{\sum_{\mathbf{c} \in \mathbb{C}} \mathbf{P}(\mathbf{c})} \sum_{\mathbf{c} \in \mathbb{C}} \Delta[\mathbf{c} \not\vdash M]\, \mathbf{P}(\mathbf{c}).$$

Find *A*: Hard Problem!

Multiplicity Assignment Problem

- KV Simplificaton: $\mathbf{x} \rightarrow \Pi \stackrel{A}{\rightarrow} M$

Multiplicity Assignment Problem

- KV Simplificaton: $\mathbf{x} \to \Pi \overset{A}{\to} M$
- Independence Assumption: $\mathbf{P}(\mathbf{x}) = \prod_{i=1}^{n} \pi_i(x_i)$

$$\mathcal{P}(\Pi, M) \triangleq \sum_{\mathbf{x} \in F^n} \Delta\left[\mathbf{x} \not\vdash M\right] \mathbf{P}(\mathbf{x})$$

Multiplicity Assignment Problem

- KV Simplificaton: $\mathbf{x} \rightarrow \Pi \stackrel{A}{\rightarrow} M$
- Independence Assumption: $\mathbf{P}(\mathbf{x}) = \prod_{i=1}^{n} \pi_i(x_i)$

$$\mathcal{P}(\Pi, M) \stackrel{\Delta}{=} \sum_{\mathbf{x} \in F^n} \Delta\left[\mathbf{x} \not\vdash M\right] \mathbf{P}(\mathbf{x})$$

- ASD Decoder:

$$\begin{aligned} P(\Pi, \gamma) &= \min_{|M| \leq \gamma} \mathcal{P}(\Pi, M) \\ M(\Pi, \gamma) &= \arg_M \min_{|M| \leq \gamma} \mathcal{P}(\Pi, M) \\ P(\Pi, \infty) &\stackrel{\Delta}{=} \lim_{\gamma \rightarrow \infty} P(\Pi, \gamma) \end{aligned}$$

Multiplicity Assignment Problem

- KV Simplificaton: $\mathbf{x} \to \Pi \xrightarrow{A} M$
- Independence Assumption: $\mathbf{P}(\mathbf{x}) = \prod_{i=1}^{n} \pi_i(x_i)$

$$\mathcal{P}(\Pi, M) \stackrel{\Delta}{=} \sum_{\mathbf{x} \in F^n} \Delta\left[\mathbf{x} \not\vdash M\right] \mathbf{P}(\mathbf{x})$$

- ASD Decoder: Hard Problem!

$$\begin{aligned} P(\Pi, \gamma) &= \min_{|M| \le \gamma} \mathcal{P}(\Pi, M) \\ M(\Pi, \gamma) &= \arg_M \min_{|M| \le \gamma} \mathcal{P}(\Pi, M) \\ P(\Pi, \infty) &\stackrel{\Delta}{=} \lim_{\gamma \to \infty} P(\Pi, \gamma) \end{aligned}$$

Soft Multiplicity Matrices

- Relax Integer Constraint: $Q = (q_i(\beta))$ 'soft' matrix.

Soft Multiplicity Matrices

- Relax Integer Constraint: $Q = (q_i(\beta))$ 'soft' matrix.
- Relaxed problem:

$$
\begin{aligned}
\mathcal{P}(\Pi, Q) &\triangleq \sum_{\mathbf{x} \in F^n} \Delta\left[\mathbf{x} \not\vdash Q\right] \mathbf{P}(\mathbf{x}) \\
P^*(\Pi, \gamma) &\triangleq \min_{|Q| \leq \gamma} \mathcal{P}(\Pi, Q) \\
Q^*(\Pi, \gamma) &\triangleq \arg \min_{|Q| \leq \gamma} \mathcal{P}(\Pi, Q) \\
P^*(\Pi, \infty) &\triangleq \lim_{\gamma \to \infty} P^*(\Pi, \gamma).
\end{aligned}
$$

Soft Multiplicity Matrices

- $\text{Theorem}: P^*(\Pi, \infty) = P(\Pi, \infty)$

Soft Multiplicity Matrices

- $\text{Theorem}: P^*(\Pi, \infty) = P(\Pi, \infty)$

$\boxed{Q^* = ? \quad \text{Still a Hard Problem?}}$

Soft Multiplicity Matrices

- $\text{Theorem} : P^*(\Pi, \infty) = P(\Pi, \infty)$

$Q^* = ?$ Still a Hard Problem?

- Minimize the Chernoff Bound on the Error Probability!

The Chernoff Bound-Finite Cost

- $\mathcal{S}_i$ are independent r.v.;

$$\langle \mathbf{x}, Q \rangle = S_Q = \mathcal{S}_1 + \cdots + \mathcal{S}_n.$$

- $$\phi_i(s, \pi_i, q_i) = E_{\mathcal{S}_i}\left\{ e^{s\mathcal{S}_i} \right\} = \sum_{\beta \in F} \pi_i(\beta) e^{s q_i(\beta)}.$$

- $$\Phi(s, \Pi, Q) = E_{S_Q}\left\{ e^{s \sum_{i=1}^{n} \mathcal{S}_i} \right\} = \prod_{i=1}^{n} \phi_i(s, \pi_i, q_i).$$

- Chernoff Bound:

$$\Pr\left\{ S_Q \le \delta \right\} \le \min_{s \ge 0} \left\{ e^{s\delta} \Phi(-s, \Pi, Q) \right\}.$$

The Chernoff Bound-Finite Cost

- Transformation [PV03]

$$X_i(\beta) = q_i(\beta) + 1/2; \quad L^2 = 2\gamma + \frac{nq}{4}; \quad D' = D_v(\gamma) + \frac{n}{2}.$$

The Chernoff Bound-Finite Cost

- Transformation [PV03]

$$X_i(\beta) = q_i(\beta) + 1/2; \quad L^2 = 2\gamma + \frac{nq}{4}; \quad D' = D_v(\gamma) + \frac{n}{2}.$$

- Transformed Problem:

$$P^*(\Pi, \gamma) \leq \min_{\|\mathbb{X}\|^2 = L^2} \min_{s \geq 0} \left\{ e^{sD'} \Phi(-s, \Pi, \mathbb{X}) \right\}.$$

- The (sub)optimum matrix

$$\mathbb{X}^* = \arg_{\mathbb{X}} \min_{\|\mathbb{X}\|^2 = L^2} \min_{s \geq 0} \left\{ e^{sD'} \Phi(-s, \Pi, \mathbb{X}) \right\}.$$

Chernoff Bound-Infinite Cost

- Theorem :$(v = k - 1)$

$$P(\Pi, \infty) = \min_{\|R\|^2=1} \sum_{\mathbf{x} \in F^n} \Delta \left[\langle \mathbf{x}, R \rangle \le \sqrt{v} \right] \mathbf{P}(\mathbf{x})$$

Chernoff Bound-Infinite Cost

- Theorem :$(v = k - 1)$

$$P(\Pi, \infty) = \min_{\|R\|^2=1} \sum_{\mathbf{x} \in F^n} \Delta \left[\langle \mathbf{x}, R \rangle \leq \sqrt{v} \right] \mathbf{P}(\mathbf{x})$$

- Chernoff Bound:

$$P(\Pi, \infty) \leq \min_{\|R\|^2=1} \min_{s \geq 0} \left\{ \Phi(-s, \Pi, R) e^{s\sqrt{v}} \right\}$$

- (Sub)Optimum Matrix:

$$R^{\chi}(\Pi) = \arg_R \min_{\|R\|^2=1} \min_{s \geq 0} \left\{ \Phi(-s, \Pi, R) e^{s\sqrt{v}} \right\}$$

The Lagrangian

- Constrained Optimization Problem:

$$\min\left(sD' + \sum_{i=1}^{n} \ln \phi_i(-s, \pi_i, X_i)\right)$$

subject to

$$s \geq 0$$

$$\|\mathbb{X}\|^2 = L^2 = 2\gamma + \frac{1}{4}nq.$$

The Lagrangian

- Constrained Optimization Problem:

$$\min \left(sD' + \sum_{i=1}^{n} \ln \phi_i(-s, \pi_i, X_i) \right)$$

subject to

$$s \geq 0$$

$$\|\mathbb{X}\|^2 = L^2 = 2\gamma + \frac{1}{4}nq.$$

- The Lagrangian:

$$\mathcal{L}(s, \mathbb{X}, \lambda) = sD' + \sum_{i=1}^{n} \ln \phi_i(-s, \pi_i, X_i) + \frac{\lambda}{2}\left(\|\mathbb{X}\|^2 - L^2\right).$$

The Lagrangian

- $\frac{\partial \mathcal{L}}{\partial \lambda}\Big|_{\lambda=\lambda^*} = 0 \;\Rightarrow\; \|\mathbb{X}\|^2 = L^2$

The Lagrangian

- $\frac{\partial \mathcal{L}}{\partial \lambda}\Big|_{\lambda=\lambda^*} = 0 \;\Rightarrow\; \|\mathbb{X}\|^2 = L^2$
-

$$D' - \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right)\Bigg|_{s=s^*} = 0$$

The Lagrangian

- $\frac{\partial \mathcal{L}}{\partial \lambda}\Big|_{\lambda=\lambda^*} = 0 \;\Rightarrow\; \|\mathbb{X}\|^2 = L^2$
- $$D' - \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right)\Bigg|_{s=s^*} = 0$$
- $$\frac{X_i(\beta)}{L^2} \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right) - \frac{\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)}\Bigg|_{\mathbb{X}=\mathbb{X}^*}$$

The Lagrangian

- $\frac{\partial \mathcal{L}}{\partial \lambda}\Big|_{\lambda=\lambda^*} = 0 \;\Rightarrow\; \|\mathbb{X}\|^2 = L^2$
- $$D' - \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right)\Bigg|_{s=s^*} = 0$$
- $$\frac{X_i(\beta)}{L^2} \sum_{i=1}^{n} \left(\frac{\sum_{\beta \in F} X_i(\beta)\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)} \right) - \frac{\pi_i(\beta)e^{-sX_i(\beta)}}{\phi_i(-s, \pi_i, X_i)}\Bigg|_{\mathbb{X}=\mathbb{X}^*}$$
- $$\frac{D'}{L^2} X_i(\beta) - \frac{\pi_i(\beta)e^{-s^* X_i(\beta)}}{\sum_{\beta \in F} \pi_i(\beta)e^{-s^* X_i(\beta)}}\Bigg|_{\mathbb{X}=\mathbb{X}^*, s=s^*} = 0$$

Convexity

- Define

$$\mathcal{L}_s(s) = \mathcal{L}^*(s, \mathbb{X})\big|_{\mathbb{X}=const}\,, \quad \mathcal{L}_{\mathbb{X}}(\mathbb{X}) = \mathcal{L}^*(s, \mathbb{X})\big|_{s=s^*}\,.$$

Convexity

- Define

$$\mathcal{L}_s(s) = \mathcal{L}^*(s, \mathbb{X})\big|_{\mathbb{X}=const}\,, \quad \mathcal{L}_{\mathbb{X}}(\mathbb{X}) = \mathcal{L}^*(s, \mathbb{X})\big|_{s=s^*}\,.$$

- $\mathcal{L}_s(s)$ is convex in s

Convexity

- Define

$$\mathcal{L}_s(s) = \mathcal{L}^*(s, \mathbb{X})\big|_{\mathbb{X}=const}\,, \quad \mathcal{L}_{\mathbb{X}}(\mathbb{X}) = \mathcal{L}^*(s, \mathbb{X})\big|_{s=s^*}\,.$$

- $\mathcal{L}_s(s)$ is convex in s
- $\mathcal{L}_{\mathbb{X}}(\mathbb{X})$ is convex in $\mathbb{X}$

Iterative Algorithm

Initialize $\mathbb{X}^o = \frac{L^2}{D'}\Pi, \quad s^o = 0.1 * \frac{D'}{L^2}$ and $j = 0.$

Do

$j := j + 1$

I. **Solve** for s^j,

$$\nabla_s(\mathcal{L}^*(s, \mathbb{X}^{j-1}) = \left.\frac{\partial \mathcal{L}^*(s, \mathbb{X}^{j-1})}{\partial s}\right|_{s=s^j} = 0$$

II. **Solve** for $\mathbb{X}^j$,

$$\nabla_{\mathbb{X}}(\mathcal{L}^*(s^j, \mathbb{X})) = \left.\left\{\frac{\partial \mathcal{L}^*(s^j, \mathbb{X})}{\partial X_i^j(\beta)}, i = 1, ..., n, \ \beta \in F\right\}\right|_{\mathbb{X}=\mathbb{X}^j} = 0$$

While $\left\|\frac{s^j - s^{j-1}}{s^{j-1}}\right\|_1 \leq \epsilon.$

Iterative Algorithm

- For finite cost multiplicity matrix $M = (m_i(\beta))$:

$$m_i(\beta) = \mathrm{Round}\left\{\max\left\{0, X_i^*(\beta) - 0.5\right\}\right\}.$$

Iterative Algorithm

- For finite cost multiplicity matrix $M = (m_i(\beta))$:

$$m_i(\beta) = \mathrm{Round}\left\{\max\left\{0, X_i^*(\beta) - 0.5\right\}\right\}.$$

- **Solve** could be replace by a Newton type algorithm.

Iterative Algorithm

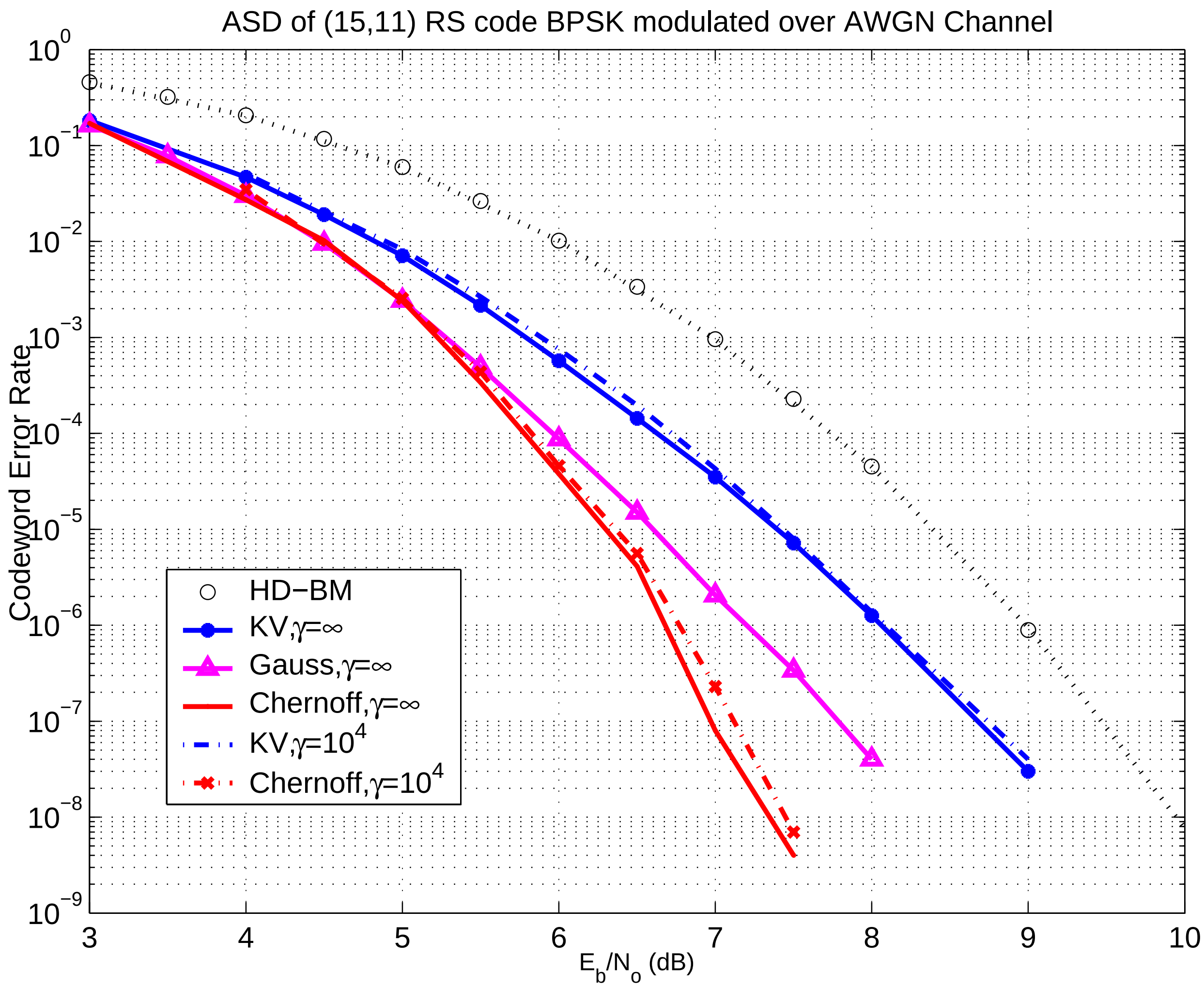
- For finite cost multiplicity matrix $M = (m_i(\beta))$:

$$m_i(\beta) = \mathrm{Round}\left\{\max\left\{0, X_i^*(\beta) - 0.5\right\}\right\}.$$

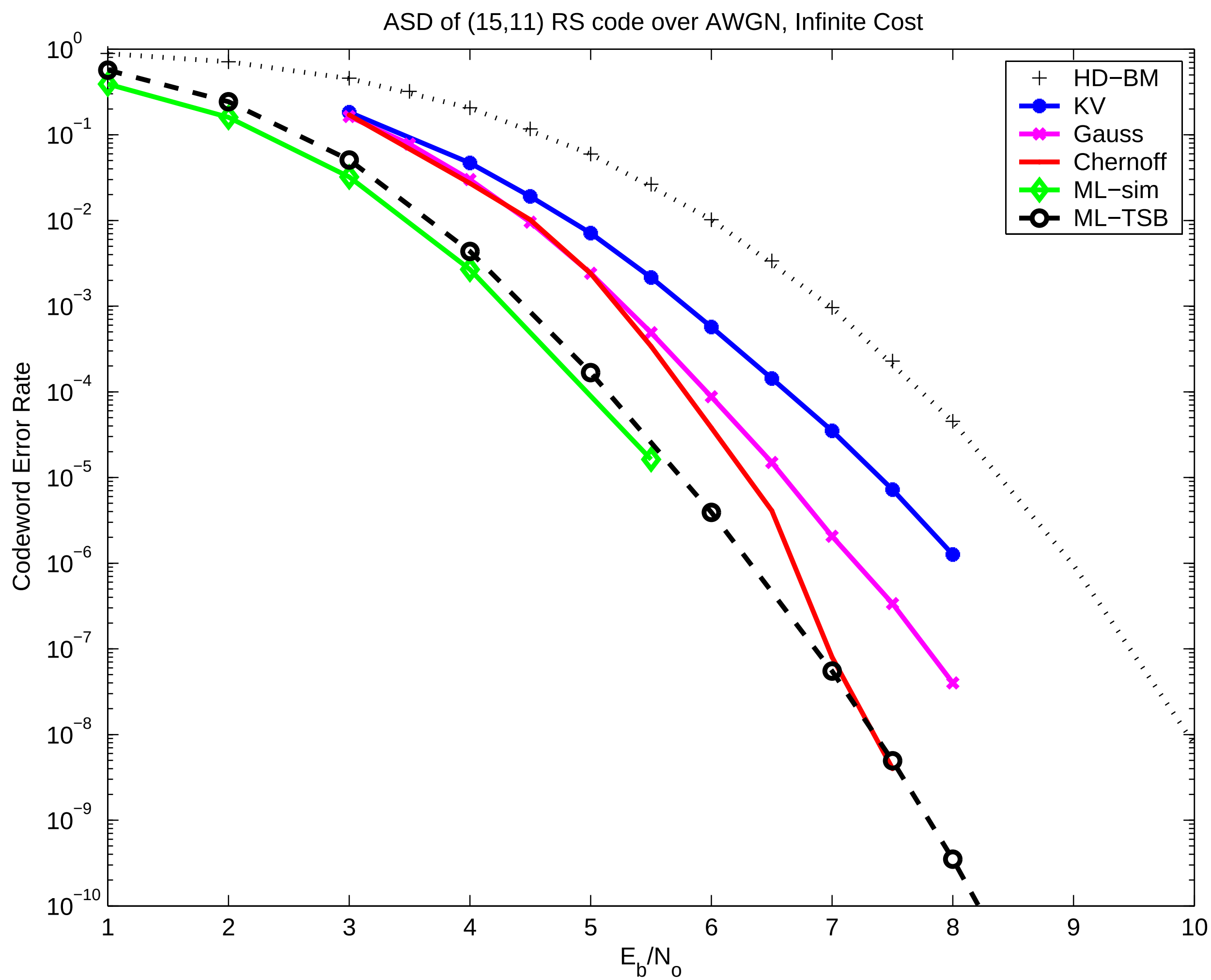
- **Solve** could be replace by a Newton type algorithm.
- To reduce computational complexity:

$$\text{Set } \; X_i(\beta) = 0 \; \text{ if } \; m_i(\beta) < \mathit{threshold}$$

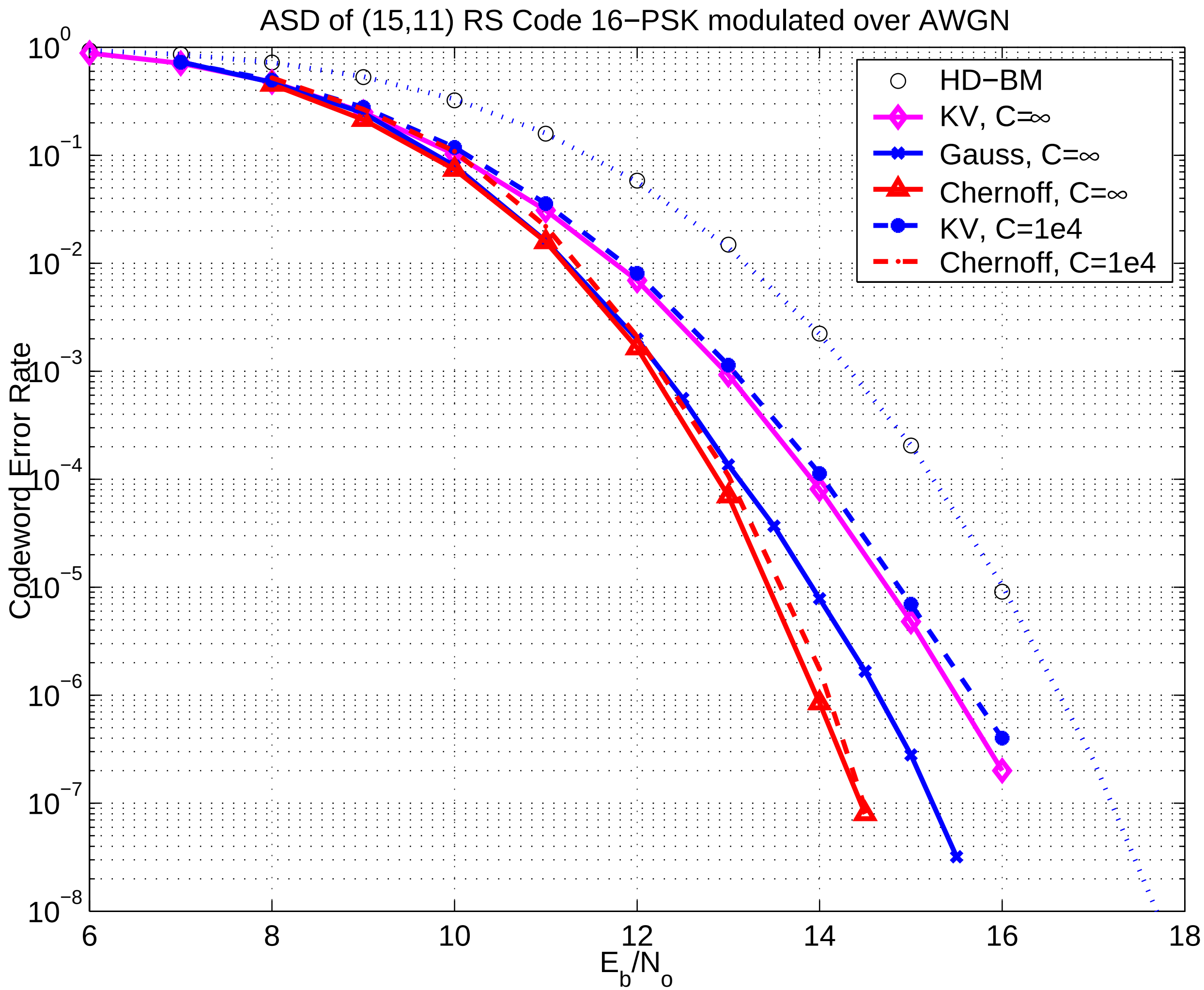
Numerical Results & Conclusions



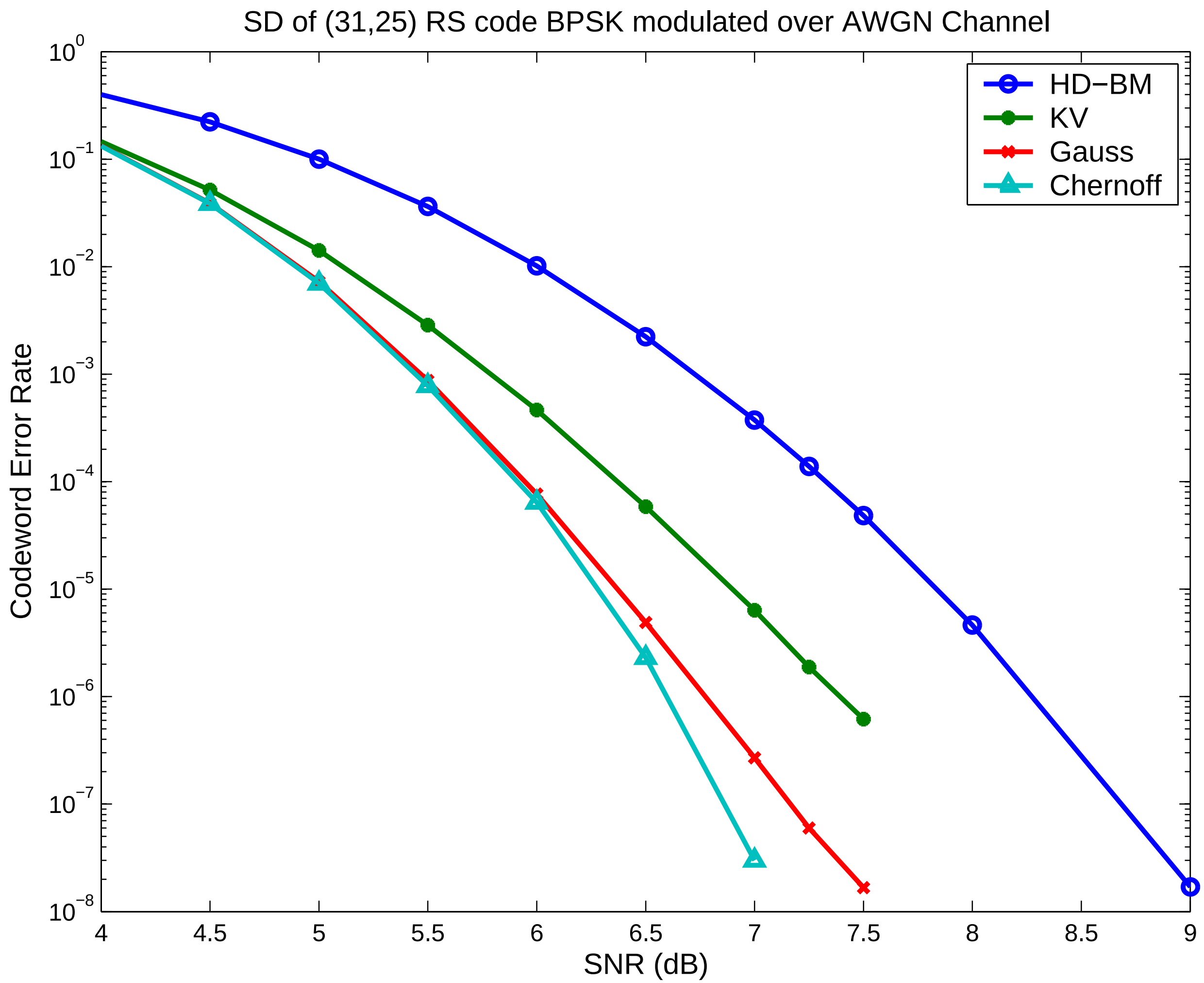
Numerical Results & Conclusions



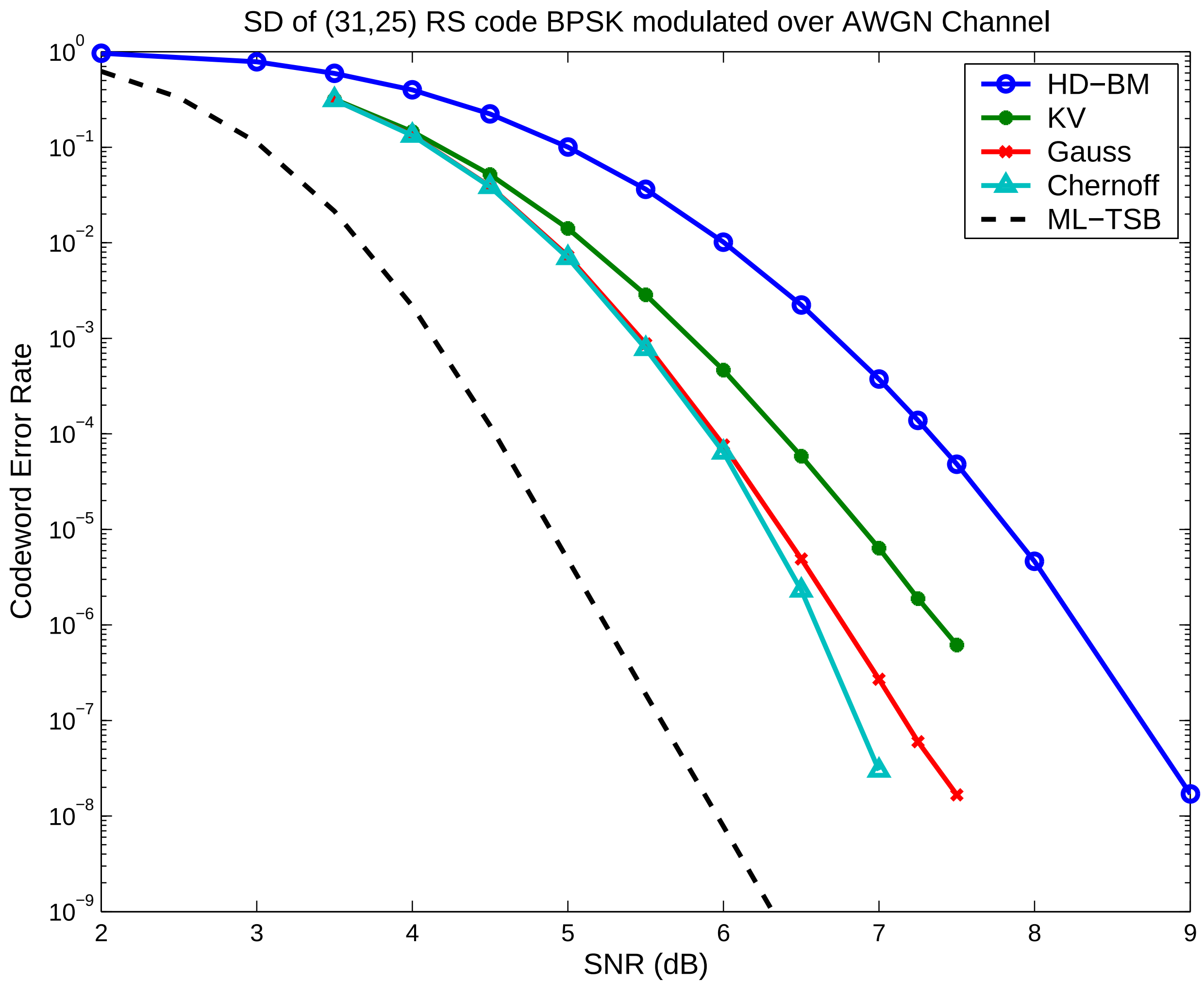
Numerical Results & Conclusions



Numerical Results & Conclusions



Numerical Results & Conclusions



Future Work

- Finding less complex Algorithms with high efficiency.
- Minimize the true Error Probability directly.
- Precondition Π to have information from other symbols and the channel.
- Iterative Algebraic Decoding.

http://mostafa.caltech.edu/Academia.html

Thank You!