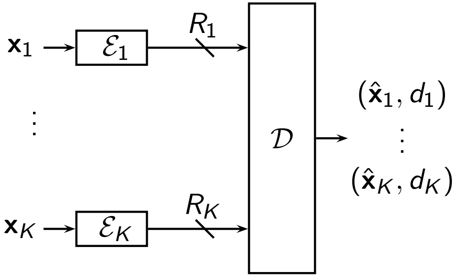
Bounds for Integer-Forcing Source Coding

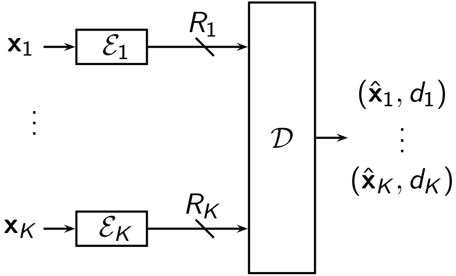
Elad Domanovitz
Joint work with Uri Erez

November 10th, 2017
ITW, Kaohsiung, Taiwan

Distributed Lossy Compression

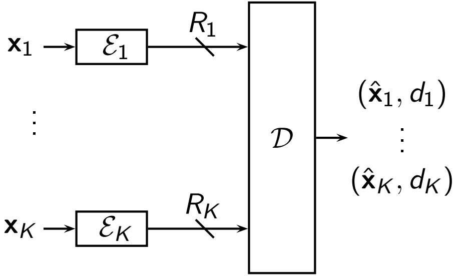


Distributed Lossy Compression



- Fundamental limits understood in some cases
- Inner and outer bounds known

Distributed Lossy Compression

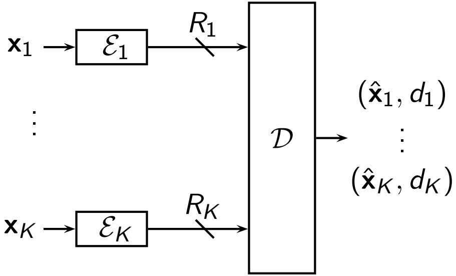


- Fundamental limits understood in some cases
- Inner and outer bounds known

Some applications require

- Extremely simple encoders/decoder
- Extremely short delay

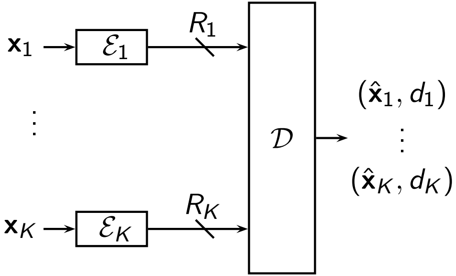
Distributed Lossy Compression



We restrict attention to:

- Gaussian sources $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$
- One-shot compression - block length is 1
- MSE distortion measure: $E(x_k - \hat{x}_k)^2 \leq d_k$

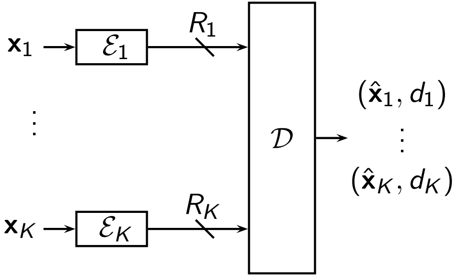
Distributed Lossy Compression



Known bounds

- An achievable rate region is Berger and Tung
- Gaussian sources - reduces to P2P quantizer + Slepian-Wolf encoding
- Optimal for two Gaussian source [Wagner '2008]

Distributed Lossy Compression



Towards a practical scheme: successive Wyner-Ziv

- Equal rates $\Longrightarrow$ Non equal distortion
- Equal distortion $\Longrightarrow$ Non equal rates
- Well understood for large blocklengths, less so for short blocks

Goal & Outline

- We're interested in a symmetric scheme:
  - Symmetric rates $R_1 = \cdots = R_K = R$
  - Symmetric distortion $d_1 = \cdots = d_K = d$
- Best known achievable scheme - Berger Tung
- A simple choice for the auxiliary random variable in BT gives: $R_{\mathrm{BT}} = \frac{1}{2} \log\det\left(\mathbf{I} + \frac{1}{d}\mathbf{K}_{xx}\right)$

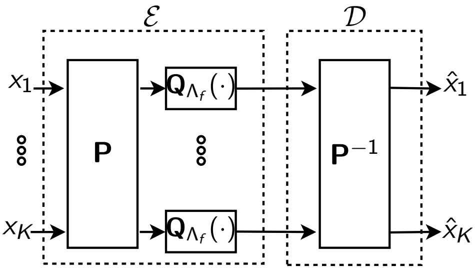
- We start by recalling **integer forcing source coding**: a simple and symmetric scheme
- Derive new bounds on **outage probability of precoded source coding integer forcing**

Motivation

- How can we utilize correlation via linear processing to reduce quantization problem to a scalar problem?

Motivation

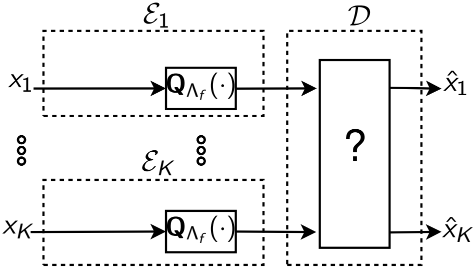
- How can we utilize correlation via linear processing to reduce quantization problem to a scalar problem?



- But diagonalization requires linear processing at both ends...

Motivation

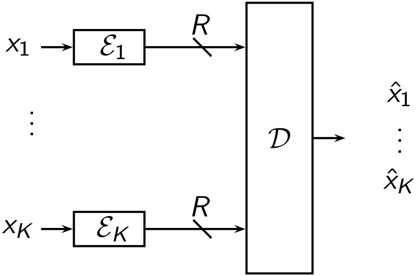
- How can we utilize correlation via linear processing to reduce quantization problem to a scalar problem?



- What can be done in case of distributed compression?

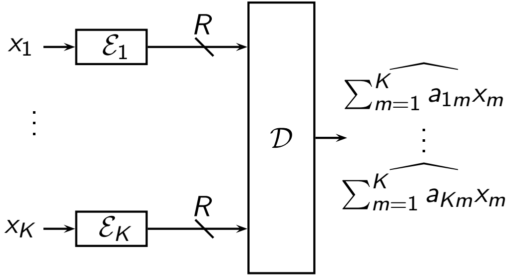
Integer-Forcing Source Coding: Overview

Basic Idea: Rather than solving the problem



Integer-Forcing Source Coding: Overview

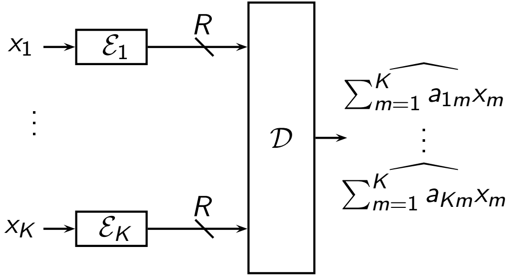
First solve



and then invert equations to get $\hat{x}_1, \ldots, \hat{x}_K$

Integer-Forcing Source Coding: Overview

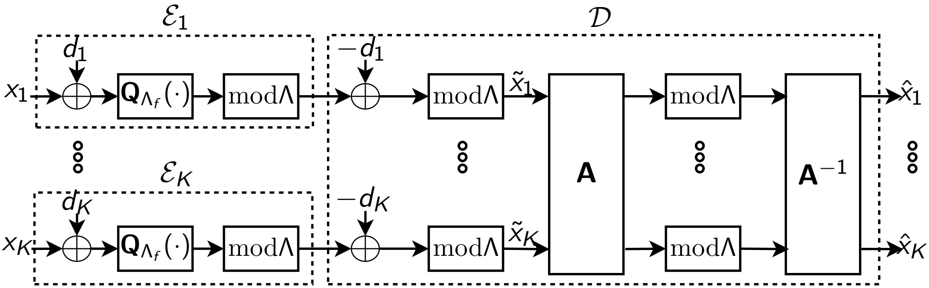
First solve



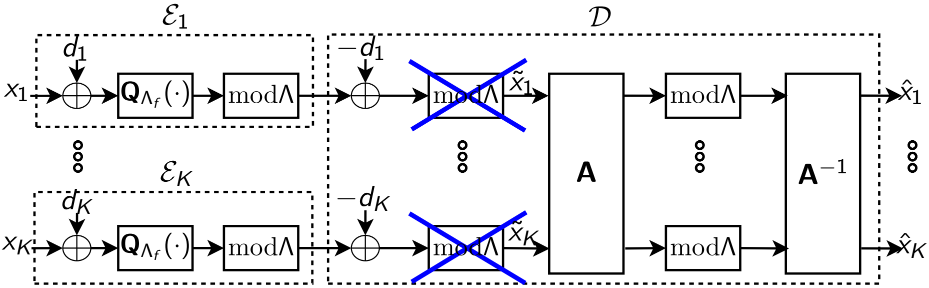
and then invert equations to get $\hat{x}_1, \ldots, \hat{x}_K$

- Problem reduces to simultaneous distributed compression of K linear combinations
- Can be efficiently solved with small rates for certain choices of coefficients
- Equation coefficients can be chosen to optimize performance

Integer Forcing Source Coding: Block Diagram

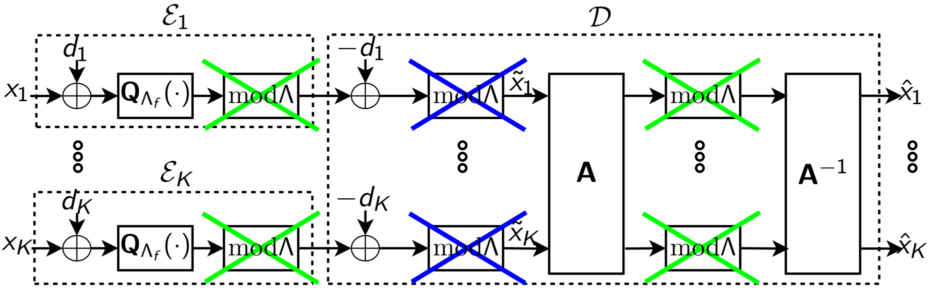


Integer Forcing Source Coding: Block Diagram



Modulo operation after dither reduction is redundant

Integer Forcing Source Coding: Block Diagram



Cancelled due to simple modulo property and careful choice of A

Explained in the following slides

Distributed Compression of Integer Linear Combination

Encoders

Each encoder is a modulo scalar quantizer with rate R : produces $\tilde{x}_k^*$

Distributed Compression of Integer Linear Combination

Encoders

Each encoder is a modulo scalar quantizer with rate R : produces $\tilde{x}_k^*$

Simple modulo property (main point)

For any set of integers $a_1, \ldots, a_K$ and real numbers $\tilde{x}_1, \ldots, \tilde{x}_K$

$$\left[\sum_{k=1}^{K} a_k \tilde{x}_k\right]^* = \left[\sum_{k=1}^{K} a_k \tilde{x}_k^*\right]^*$$

Distributed Compression of Integer Linear Combination

Encoders

Each encoder is a modulo scalar quantizer with rate R : produces $\tilde{x}_k^*$

Simple modulo property (main point)

For any set of integers $a_1, \ldots, a_K$ and real numbers $\tilde{x}_1, \ldots, \tilde{x}_K$

$$\left[\sum_{k=1}^{K} a_k \tilde{x}_k\right]^* = \left[\sum_{k=1}^{K} a_k \tilde{x}_k^*\right]^*$$

Decoder

- Gets: $\tilde{x}_1^*, \ldots, \tilde{x}_K^*$
- Outputs:

$$\widehat{\mathbf{a}^T\mathbf{x}} = \left[\sum_{k=1}^{K} a_k \tilde{x}_k^*\right]^* = \left[\sum_{k=1}^{K} a_k \tilde{x}_k\right]^* = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^*$$

Compression of Integer Linear Combination

$$\widehat{\mathbf{a}^T\mathbf{x}} = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^* = \begin{cases} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u} \\ \text{error} \end{cases}$$

What is a good $\mathbf{a}$?

For a given modulo interval Δ (not to be confused with the quantization step size), a good $\mathbf{a}$ is such that modulo is not active

- If $\frac{\mathbb{E}(\|\mathbf{a}^T(\mathbf{x}+\mathbf{u})\|)^2}{n} \leq \frac{\Delta^2}{n} \Longrightarrow \widehat{\mathbf{a}^T\mathbf{x}} \overset{\text{w.h.p}}{=} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u}$
- Small $\Delta \Longrightarrow$ small R

Compression of Integer Linear Combination

$$\widehat{\mathbf{a}^T\mathbf{x}} = \left[\mathbf{a}^T(\mathbf{x}+\mathbf{u})\right]^* = \begin{cases} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u} \\ \text{error} \end{cases}$$

What is a good a?

For a given modulo interval Δ (not to be confused with the quantization step size), a good $\mathbf{a}$ is such that modulo is not active

- If $\frac{\mathbb{E}(\|\mathbf{a}^T(\mathbf{x}+\mathbf{u})\|)^2}{n} \leq \frac{\Delta^2}{n} \implies \widehat{\mathbf{a}^T\mathbf{x}} \overset{\text{w.h.p}}{=} \mathbf{a}^T\mathbf{x} + \mathbf{a}^T\mathbf{u}$
- Small $\Delta \implies$ small R

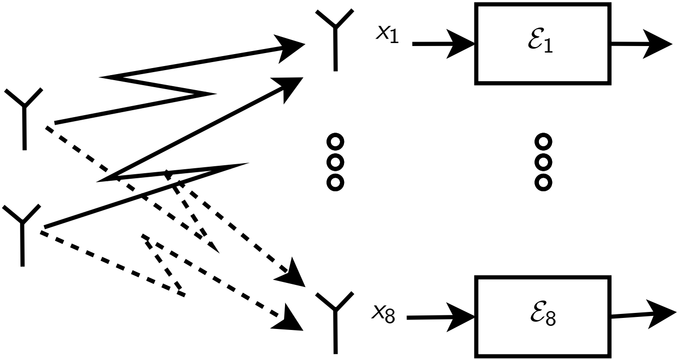
- Now, form $\mathbf{a}_1 \ldots \mathbf{a}_K$ and define $\mathbf{A} = \left[\mathbf{a}_1 \ldots \mathbf{a}_K\right]^T$

Theorem [Ordentlich '17]

$$R_{\text{IF}}(\mathbf{A}, d) \triangleq \frac{1}{2}\log\left(\max_{m=1,\ldots,K} \mathbf{a}_m^T\left(\mathbf{I} + \frac{1}{d}\mathbf{K}_{\mathbf{xx}}\right)\mathbf{a}_m\right)$$

Integer-Forcing Source Coding: Example

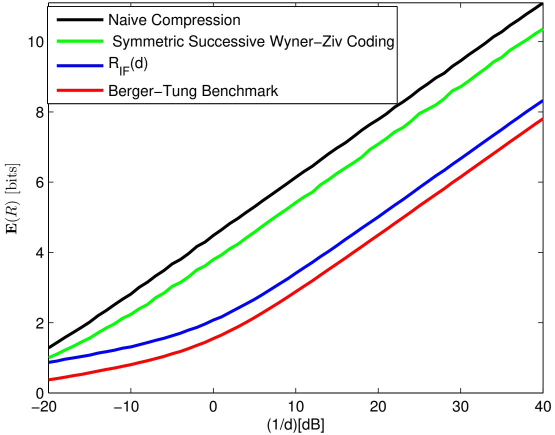
$$\mathbf{H} \in \mathbb{R}^{8\times 2}, \text{ i.i.d. Rayleigh, SNR} = 20\text{dB}$$
$$\implies \mathbf{x} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{K_{xx}}\right), \mathbf{K_{xx}} = \mathbf{I} + \text{SNR}\mathbf{H}\mathbf{H}^T$$



We show next the expected compression rate with IF source coding

Integer-Forcing Source Coding: Example

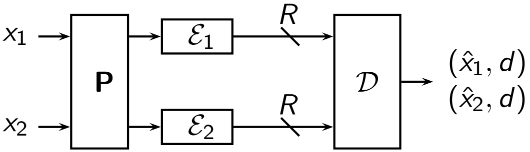
$\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{K_{xx}})$, $\mathbf{K_{xx}} = \mathbf{I} + \mathrm{SNR}\mathbf{H}\mathbf{H}^T$, SNR = 20dB and $\mathbf{H} \in \mathbb{R}^{8\times 2}$



Pleasing empirical results; what can be analyzed?

Outage Analysis of Precoded IF Source Coding

$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$



- What is outage in integer forcing source coding?
- Define the compound class of Gaussian sources with $\mathbf{K}_{xx}$ s.t.:

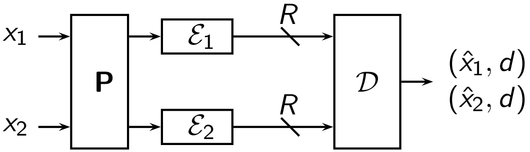
$$\mathbb{K}(R_{\mathrm{BT}}) = \left\{ \mathbf{K}_{\mathbf{xx}} \in \mathbb{R}^{K \times K} : \log\det\left(\mathbf{I} + \mathbf{K}_{\mathbf{xx}}\right) = R_{\mathrm{BT}} \right\}$$

- The worst-case (WC) scheme outage probability is defined as

$$P^{\mathrm{WC}}_{\mathrm{out,IF}}\left(R_{\mathrm{BT}}, \Delta R\right) = \sup_{\mathbf{K}_{\mathbf{xx}} \in \mathbb{K}(R_{\mathrm{BT}})} \mathrm{Pr}\left(R_{\mathrm{IF}}(\mathbf{K}_{xx}) > R_{\mathrm{BT}} + \Delta R\right)$$

Outage Analysis of Precoded IF Source Coding

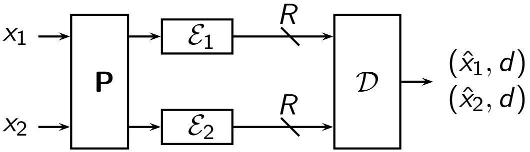
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



This is no longer distributed compression...

Outage Analysis of Precoded IF Source Coding

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



This is no longer distributed compression...

Bear with me for a few more slides

Performance of Precoded IF Source Coding

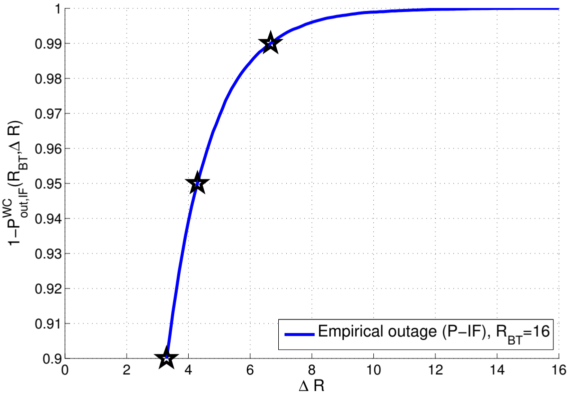
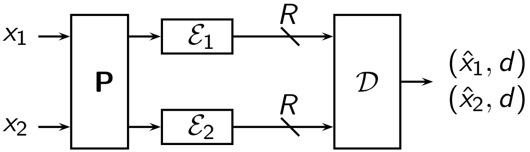


Figure: Zoom in on the empirical WC outage probability for a two-dimensional Gaussian source vector with $R_{\mathrm{BT}} = 16$ and uniform (Haar) disturbed over orthogonal matrices **P**

Outage Analysis of Precoded IF Source Coding

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



Theorem (worst-case outage of precoded IF)

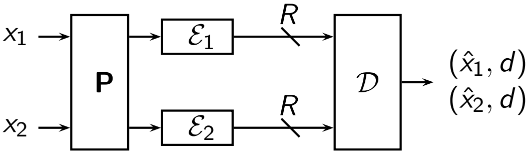
For any K sources with Berger-Tung rate of R_{BT}, and for $\mathbf{P}$ drawn from the CRE we have

$$\Pr\left(R_{\mathrm{P-IF}}(\mathbf{K}_{xx}, \mathbf{P}) > R_{\mathrm{BT}} + \Delta R\right) < c(K) 2^{-\Delta R}$$

where $c(K) = K\left(\frac{K+3}{4}\gamma_K^{\Delta^2}\right)^{\frac{K}{2}}\left(1+\sqrt{K}\right)^K \frac{\pi^{K/2}}{\Gamma(K/2+1)}$

C-RAN: CRE Precoding by Nature

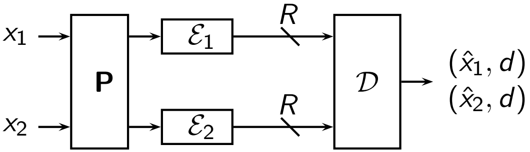
$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



This is no longer distributed compression...

C-RAN: CRE Precoding by Nature

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \mathcal{N}(\mathbf{0}, \mathbf{K}_{xx})$$



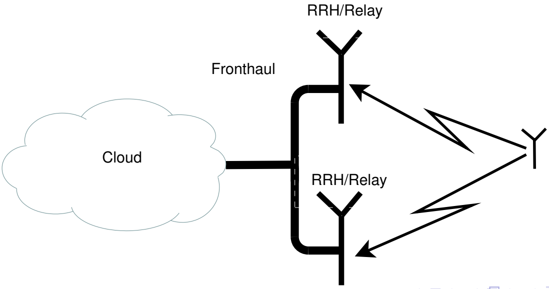
This is no longer distributed compression...

But there are cases where the precoding can be viewed as was done by nature

C-RAN: CRE Precoding by Nature

Cloud Radio Access Network (C-RAN) over Rayleigh channel

- The covariance of the received signal $\mathbf{K}_{xx} = SNR\mathbf{H}\mathbf{H}^T + \mathbf{I}$
- W.L.O.G assume $SNR = 1 \Longrightarrow \mathbf{K}_{xx} = \mathbf{H}\mathbf{H}^T + \mathbf{I}$
- $\mathbf{H} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$
- $\mathbf{K}_{xx} = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T\mathbf{U}$
- $\mathbf{H}_{i,j} \sim \mathcal{N}(0, \sigma^2)$ and i.i.d. $\Longrightarrow \mathbf{U}, \mathbf{V}^T$ are Haar distributed



C-RAN: CRE Precoding by Nature

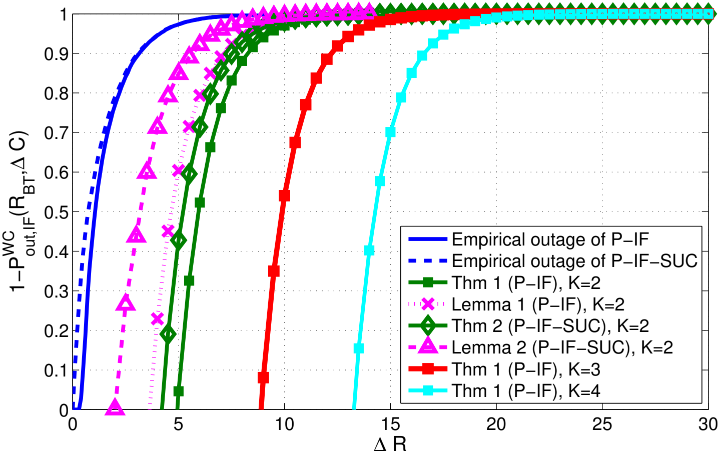


Figure: Outage bounds for different number of sources.

Thanks for your attention!