Bits and Flops in Non-Ergodic MIMO:

Can (a reasonable number of) flops provably offer ergodic-like behavior

Petros Elia

Performance and complexity in MIMO communications

- Setting of interest: general *outage limited* MIMO communications

$$\mathbf{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w}$$

  - MIMO, MIMO-OFDM, MIMO-MAC, MIMO-ARQ, COOPERATIVE, HYBRID...
  - Rx knows $\boldsymbol{H}$, Tx does not

Examples:
- Communication of CSIT over feedback link (even with reciprocity in multiuser case)
- After interference cancellation

$$\big(\text{SNR } \rho, \quad \text{rate } R, \quad \text{reliability } P_{\text{err}}, \quad \text{complexity } C\big)$$

Rate and reliability measure and exponent

$$P_{\text{err}} = P(\hat{\mathbf{x}} \neq \boldsymbol{x}_{\text{tx}})$$

$$R = \frac{1}{T} \log |\text{Code}|, \quad |\text{Code}| = 2^{RT}$$

HIGH-SNR P_{ERR} BEHAVIOR: EXPONENT OVER ρ

$$d(r) := - \lim_{\rho \to \infty} \frac{\log P_{\text{err}}}{\log \rho}, \qquad P_{\text{err}} \doteq \rho^{-d(r)} \qquad r = \frac{R}{\log \rho}$$

Computational complexity measure

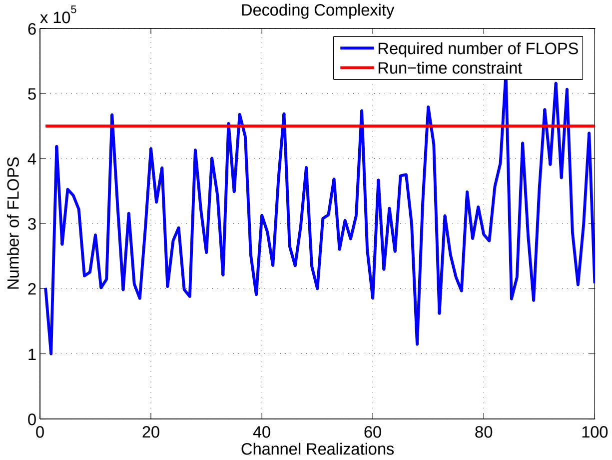
$$N_{\max}$$

MAXIMUM ALLOWABLE COMPUTATIONAL RESOURCES (PER T CHANNEL USES)

- chip size, number of flops (after that effort must terminate), etc.

Fluctuating complexity introduces a tradeoff

- Keep in mind: Generally complexity fluctuates with channel
- Generally $P_{\text{err}} \uparrow$ as $N_{\max} \downarrow$



Instantaneous algorithmic complexity fluctuations

Small example

- Can you achieve $(P_{\text{err}}, R, \rho)$ with $N_{\text{max}} = 2000$ flops?
  - ⋆ *No! Too common early-terminations for search based decoders ($N(H)$ varies) - or too weak linear receivers*
- Can you do it with $N_{\text{max}} = 100000$ flops?
  - ⋆ *No, but we are getting there.*
- How about with 132957 flops?
  - ⋆ *Yes!*
- How about with 132956 flops?
  - ⋆ *No!*
- OK, for $(P_{\text{err}}, R, \rho)$ you need $N_{\text{max}} = 132957$ flops.
  Else $(P_{\text{err}}, R, \rho)$ is not achievable.

Complexity exponent

$$c(r) := \lim_{\rho \to \infty} \frac{\log N_{\max}}{\log \rho},$$

$$N_{\max} \doteq \rho^{c(r)} = 2^{R\frac{c(r)}{r}} \dot{\leq} \rho^{rT} = |\mathcal{X}|$$

$$c(r) > 0 \implies N_{\max} \text{ exponential in } R \text{ (and often in } RT)$$

Meaningful matching of error and complexity exponents

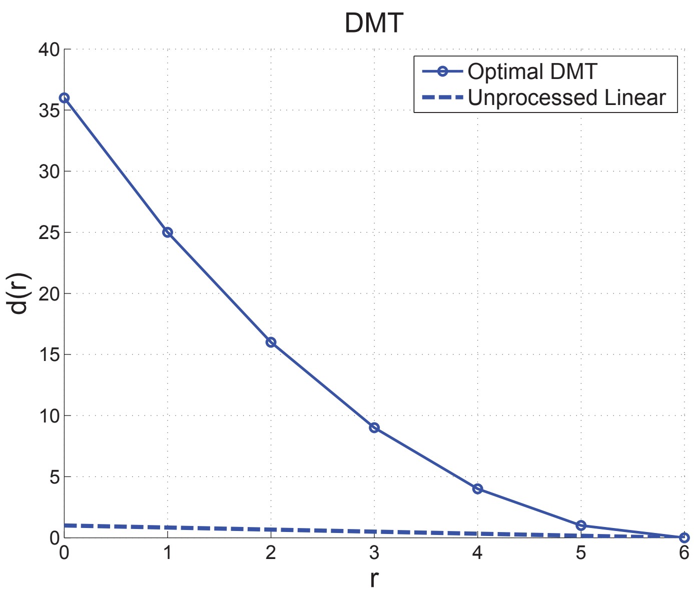
$$c(r) := \lim_{\rho\to\infty} \frac{\log N_{\max}}{\log \rho}, \qquad d(r) := -\lim_{\rho\to\infty} \frac{\log P_{\text{err}}}{\log \rho}$$

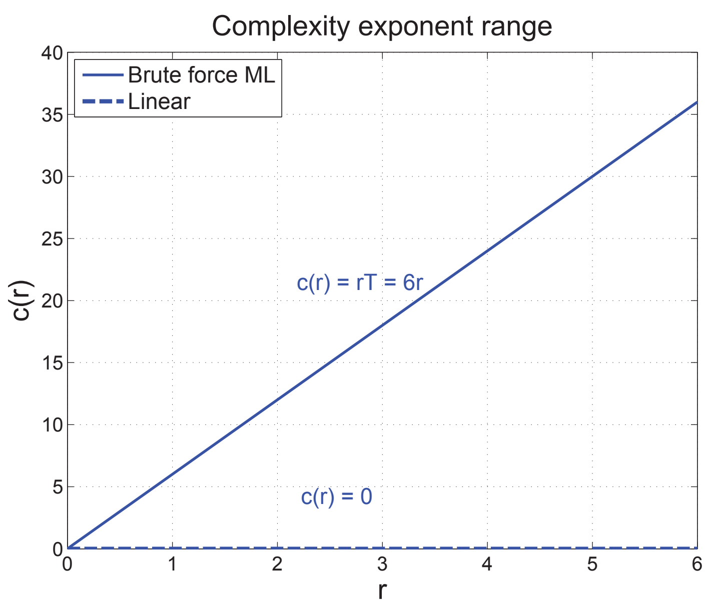
- Reliability and complexity naturally polynomial in ρ

$$N_{\max} : 1 \to K \cdot |\text{Code}| \approx 2^{RT} \approx \rho^{rT}, \qquad P_{\text{err}} : \rho^0 \to \rho^{-d_{\text{opt}}(r)}$$

Practical ramifications of both exponents

- Performance: From highly unreliable to near-ergodic reliability
- Complexity: From easy to impossible
  - ⋆ $c(r) = 0$: linear - very fast
  - ⋆ $c(r) = rT \longrightarrow N_{\max} = 2^{rT} \rightarrow \rho^{36}$





For now focus on search-based lattice decoders

- For now we neglect linear receivers
  - ⋆ Without lattice reduction they are extremely suboptimal
  - ⋆ May have unbounded gap to optimal solutions even with LR
  - ⋆ LR problematic in ubiquitous scenarios (inner-outer codes)

- We instead focus on search based (ML and lattice decoding)
  - ⋆ aka: ML sphere decoding, and lattice sphere decoding

- We also focus on linear lattice code designs

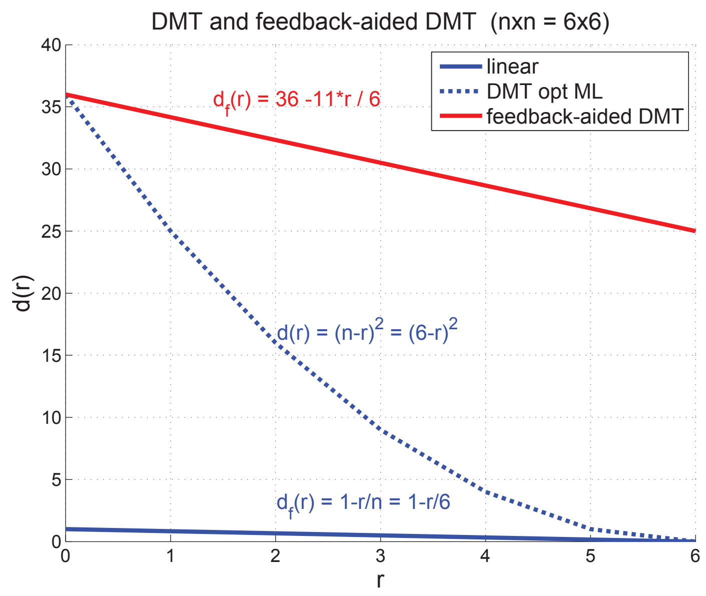
Search-based decoders

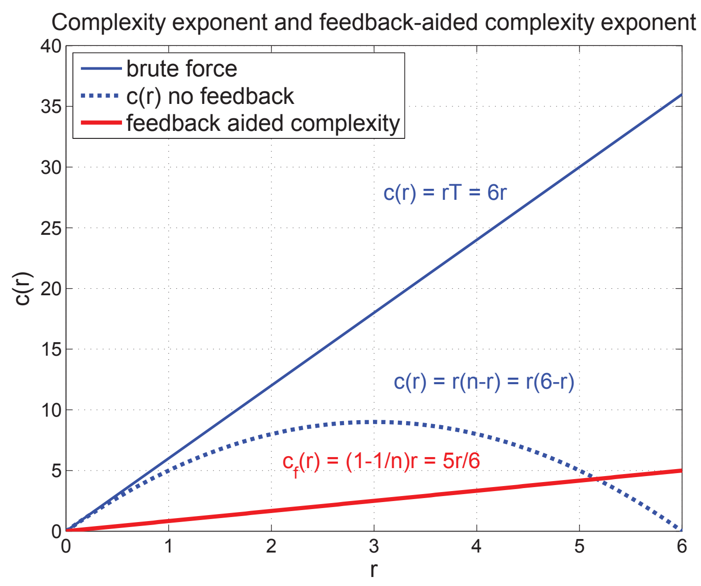
ML-based sphere decoding

(Regularized) Lattice-based sphere decoding

no lattice reduction

Preview: $d(r), c(r)$ for ML and lattice decoders





- Answer will lie somewhere in the middle

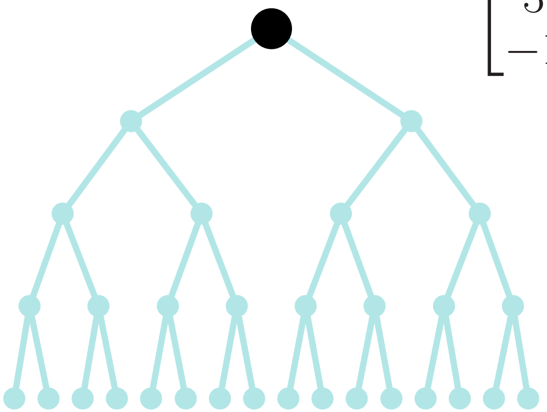
Search based decoders

$$\mathbf{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{w} = \boldsymbol{Q}\boldsymbol{R}\boldsymbol{x} + \boldsymbol{w}$$

ML: Solve $\min_{\hat{\boldsymbol{x}} \in \text{Code}} \|\tilde{\boldsymbol{y}} - \boldsymbol{R}\hat{\boldsymbol{x}}\|^2$

(Regularized) Lattice decoding: Solve $\min_{\hat{\boldsymbol{x}} \in \text{Lattice}} \|\tilde{\boldsymbol{y}} - \boldsymbol{R}\hat{\boldsymbol{x}}\|^2 + \alpha\|\hat{\boldsymbol{x}}\|_T^2$

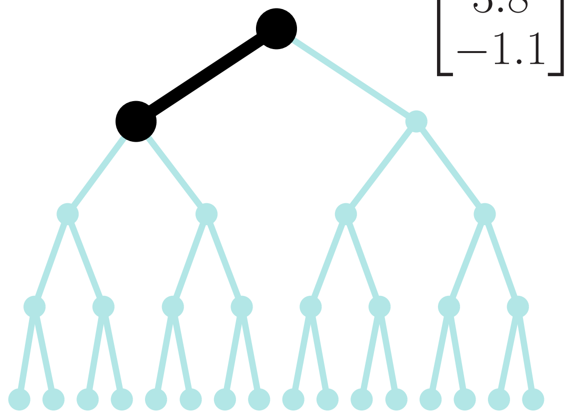
by searching over $\|\tilde{\boldsymbol{y}} - \boldsymbol{R}\hat{\boldsymbol{x}}\|^2 \leq (\text{radius})^2$



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix}$$

$$= 0.00 \leq 4.0$$

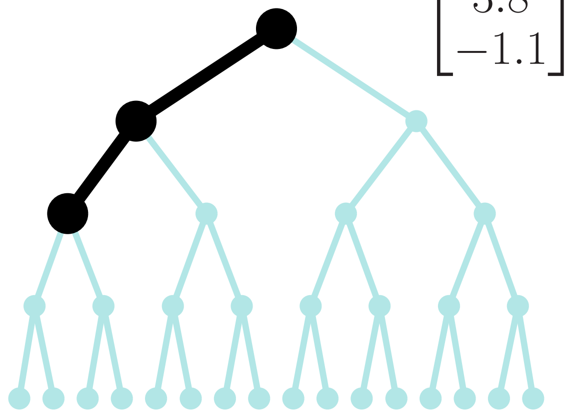
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ -0.5 \end{bmatrix}$$

$$0.5^2 = 0.25 \leq 4.0$$

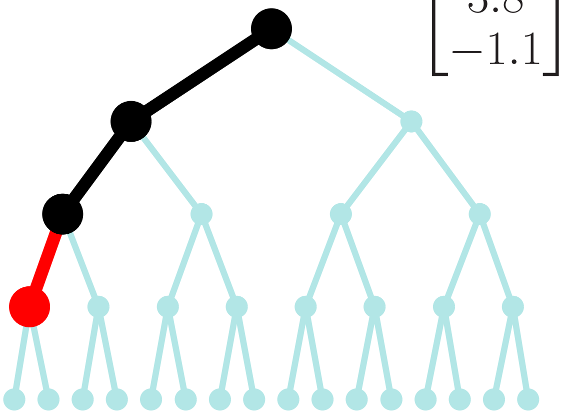
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 1.6 \\ -0.5 \end{bmatrix}$$

$$1.6^2 + 0.5^2 = 2.81 \leq 4.0$$

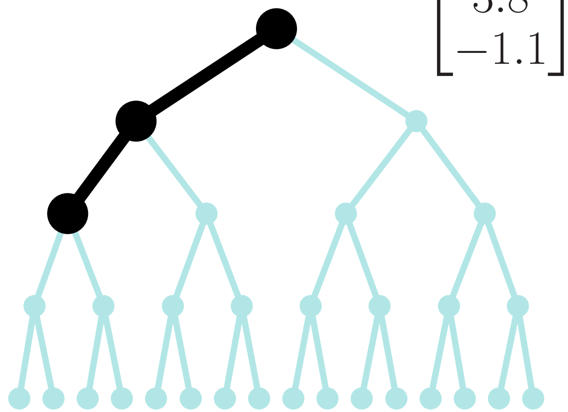
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ -1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ -3.5 \\ 1.6 \\ -0.5 \end{bmatrix}$$

$$3.5^2 + 1.6^2 + 0.5^2 = 15.06 > 4.0$$

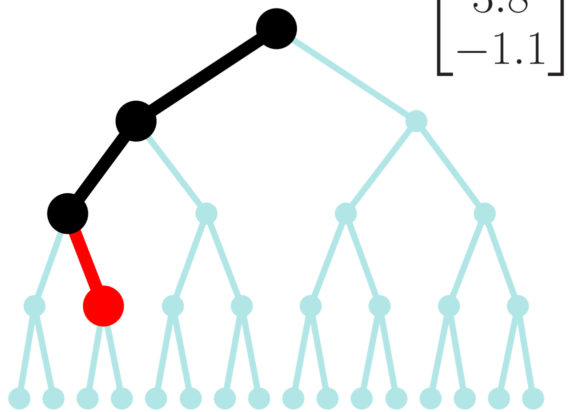
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 1.6 \\ -0.5 \end{bmatrix}$$

$$1.6^2 + 0.5^2 = 2.81 \leq 4.0$$

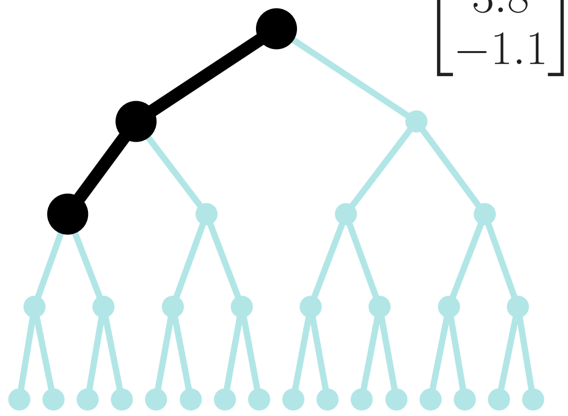
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ +1 \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ -4.3 \\ 1.6 \\ -0.5 \end{bmatrix}$$

$$4.3^2 + 1.6^2 + 0.5^2 = 21.30 > 4.0$$

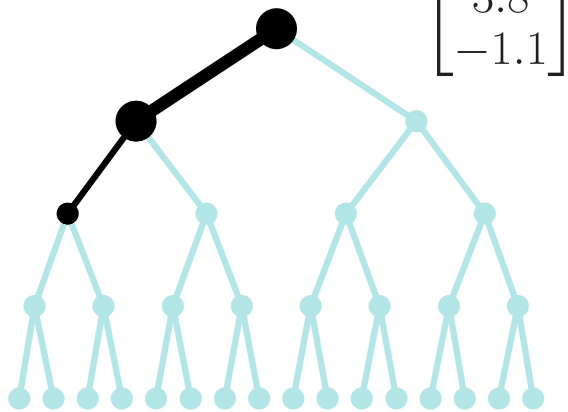
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ -1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 1.6 \\ -0.5 \end{bmatrix}$$

$$1.6^2 + 0.5^2 = 2.81 \leq 4.0$$

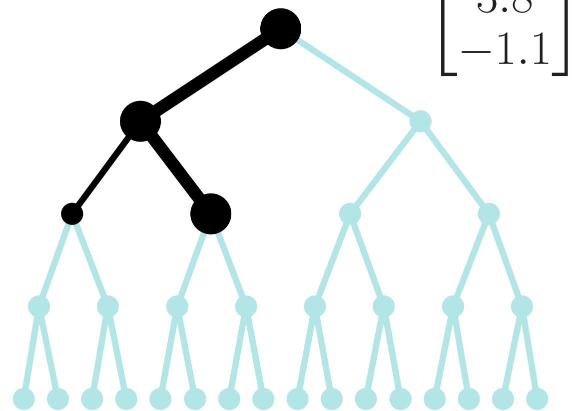
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ -0.5 \end{bmatrix}$$

$$0.5^2 = 0.25 \leq 4.0$$

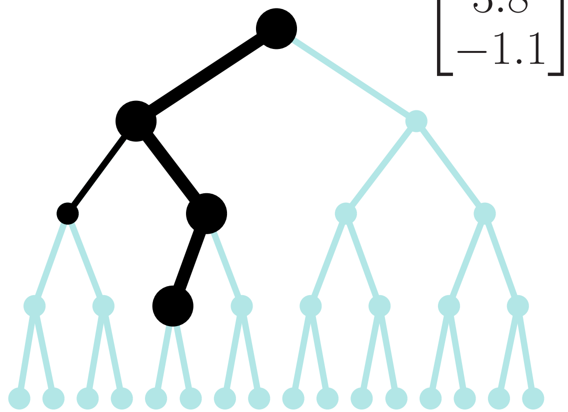
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 0.6 \\ -0.5 \end{bmatrix}$$

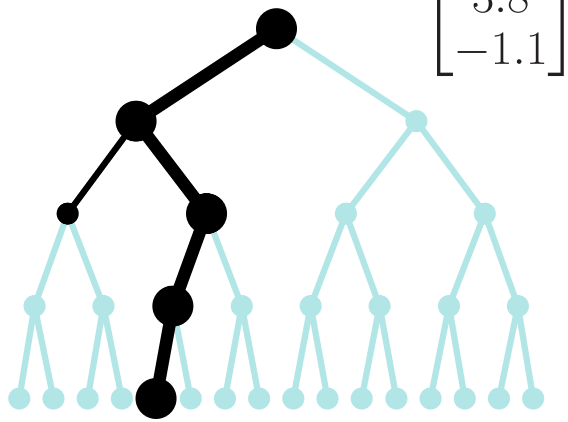
$$0.6^2 + 0.5^2 = 0.61 \leq 4.0$$

Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ -1.1 \\ 0.6 \\ -0.5 \end{bmatrix}$$

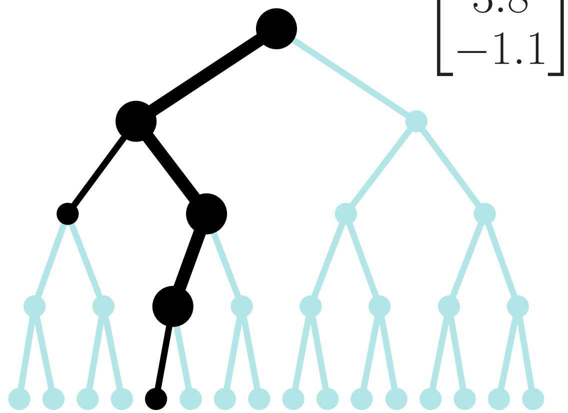
$$1.1^2 + 0.6^2 + 0.5^2 = 1.82 \leq 4.0$$



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} -1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} -0.3 \\ -1.1 \\ 0.6 \\ -0.5 \end{bmatrix}$$

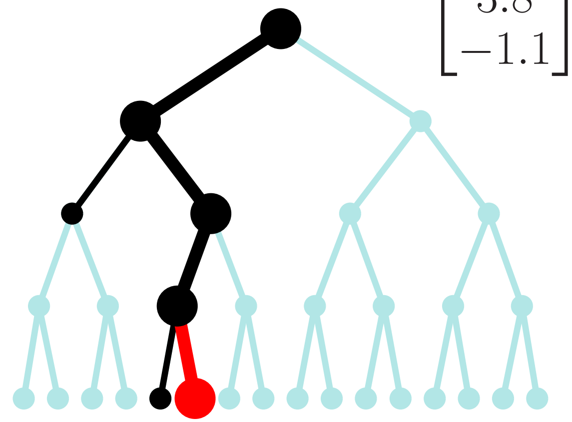
$$0.3^2 + 1.1^2 + 0.6^2 + 0.5^2 = 1.91 \leq 4.0$$

Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ -1.1 \\ 0.6 \\ -0.5 \end{bmatrix}$$

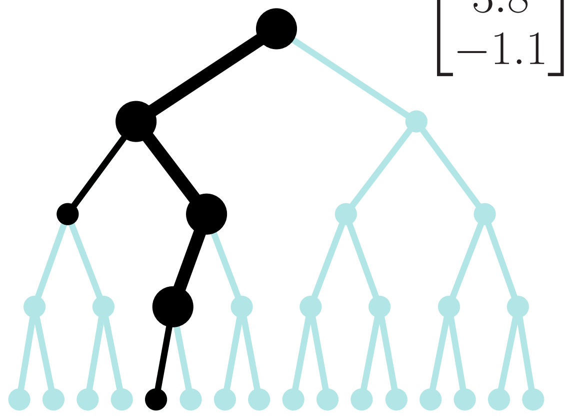
$$1.1^2 + 0.6^2 + 0.5^2 = 1.82 \leq 4.0$$



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} +1 \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} -4.5 \\ -1.1 \\ 0.6 \\ -0.5 \end{bmatrix}$$

$$4.5^2 + 1.1^2 + 0.6^2 + 0.5^2 = 22.07 > 4.0$$

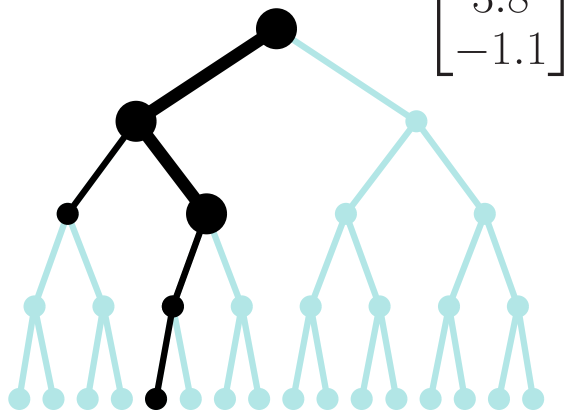
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ -1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ -1.1 \\ 0.6 \\ -0.5 \end{bmatrix}$$

$$1.1^2 + 0.6^2 + 0.5^2 = 1.82 \leq 4.0$$

Sphere decoding example, numerics

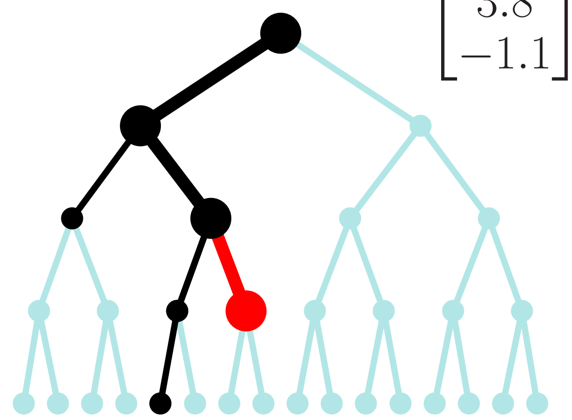


$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 0.6 \\ -0.5 \end{bmatrix}$$

$$0.6^2 + 0.5^2 = 0.61 \leq 4.0$$

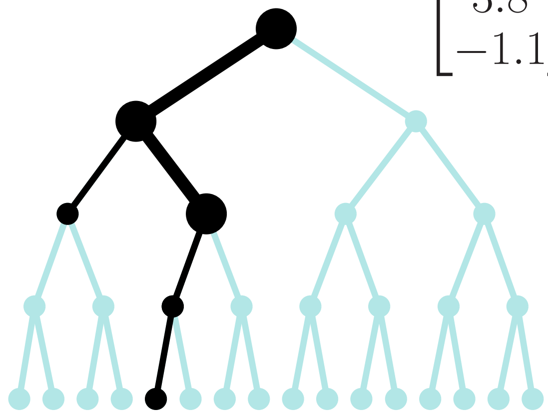
Sphere decoding example, numerics

$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ +1 \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ -1.9 \\ 0.6 \\ -0.5 \end{bmatrix}$$



$$1.9^2 + 0.6^2 + 0.5^2 = 4.22 > 4.0$$

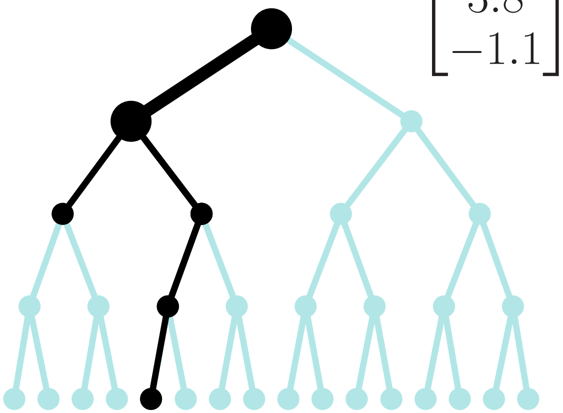
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ +1 \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 0.6 \\ -0.5 \end{bmatrix}$$

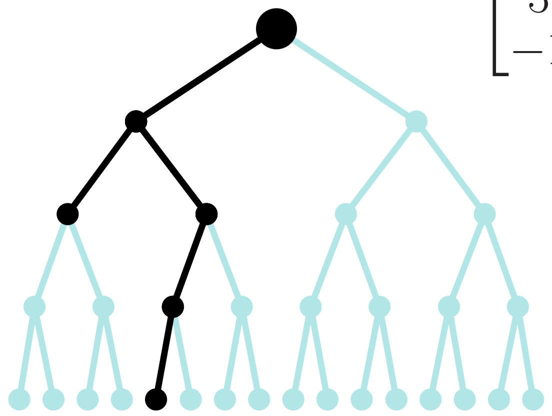
$$0.6^2 + 0.5^2 = 0.61 \leq 4.0$$

Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ -1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ -0.5 \end{bmatrix}$$

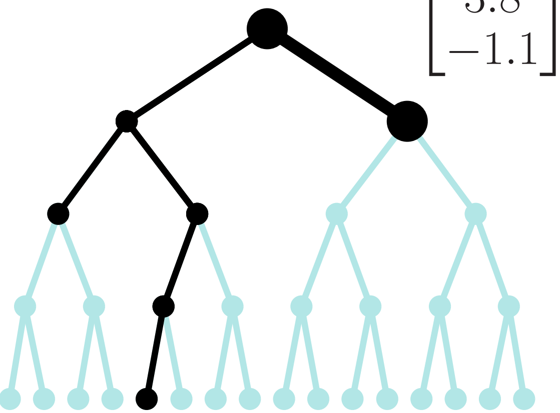
$$0.5^2 = 0.25 \leq 4.0$$



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix}$$

$$= 0.00 \leq 4.0$$

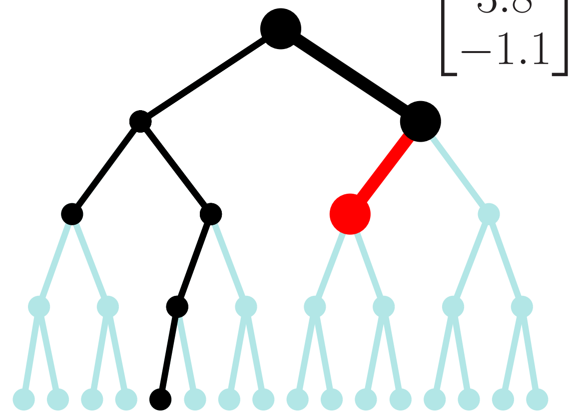
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ +1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ -1.7 \end{bmatrix}$$

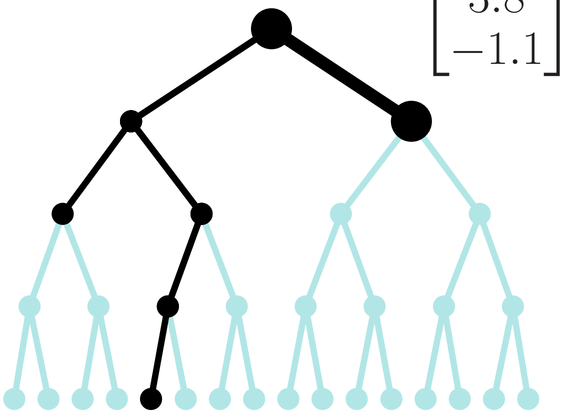
$$1.7^2 = 2.89 \leq 4.0$$

Sphere decoding example, numerics



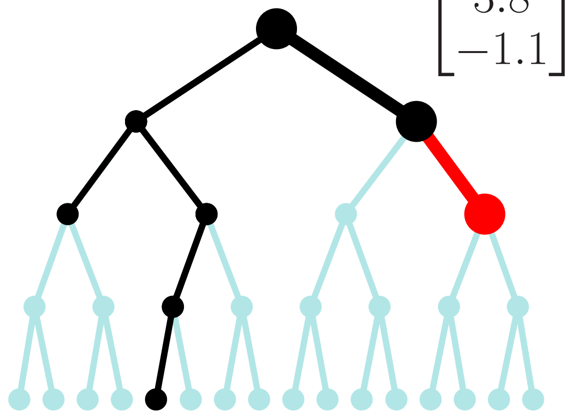
$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ -1 \\ +1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 7.0 \\ -1.7 \end{bmatrix}$$

$$7.0^2 + 1.7^2 = 51.89 > 4.0$$



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ +1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ -1.7 \end{bmatrix}$$

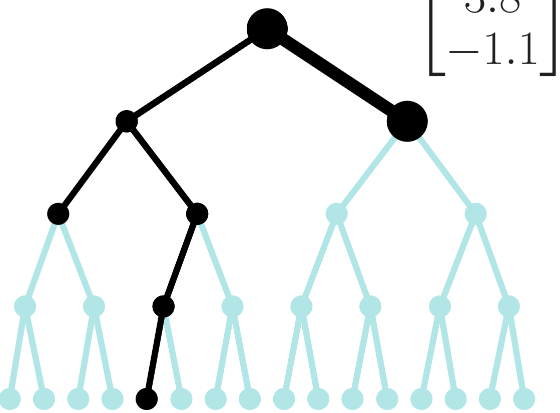
$$1.7^2 = 2.89 \leq 4.0$$



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ +1 \\ +1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ 6.0 \\ -1.7 \end{bmatrix}$$

$$6.0^2 + 1.7^2 = 38.89 > 4.0$$

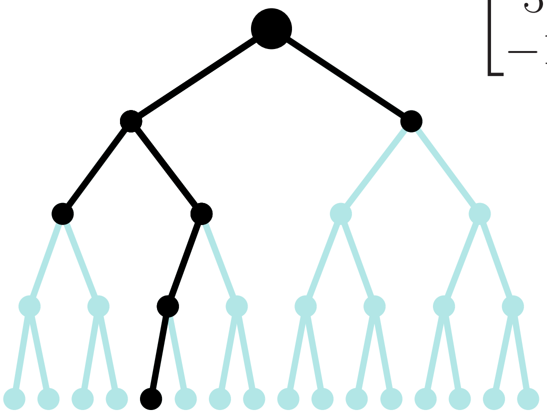
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ +1 \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ -1.7 \end{bmatrix}$$

$$1.7^2 = 2.89 \leq 4.0$$

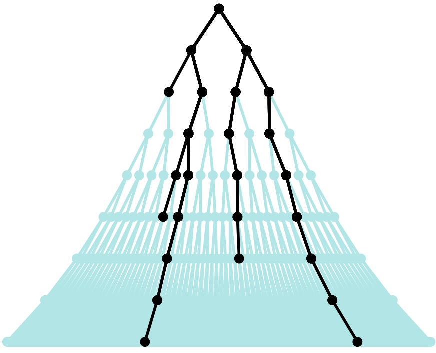
Sphere decoding example, numerics



$$\begin{bmatrix} 2.3 \\ -0.0 \\ 3.8 \\ -1.1 \end{bmatrix} - \begin{bmatrix} 2.1 & -0.9 & 1.0 & -2.8 \\ & 0.4 & -1.2 & -2.7 \\ & & 0.5 & -2.7 \\ & & & 0.6 \end{bmatrix} \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix} = \begin{bmatrix} ? \\ ? \\ ? \\ ? \end{bmatrix}$$

$$= 0.00 \leq 4.0$$

Accumulated complexity



Snapshot of instantaneous complexity

Universal bounds and equivalence of ML and lattice decoding

- We derived universal bounds for general MIMO (all scenarios, statistics, etc)
- We derived tightness whenever possible (broad setting)

- We will not get into that now: we focus on simpler more insightful settings

Theorem: *(Equivalence of ML and lattice decoding - Restatement)*
ML based sphere decoding and regularized lattice sphere decoding share the same complexity exponent for a very broad setting (share bounds and 'tightness')

$\Rightarrow$ All following results will hold for ML as well as for (regularized) lattice sphere decoding

DMT-opt quasi-static $n_T \times n_R$ ($n_T \leq n_R$)

Universal Bounds - Quasi static

Theorem: *$c(r)$ is upper bounded as (piecewise linear)*

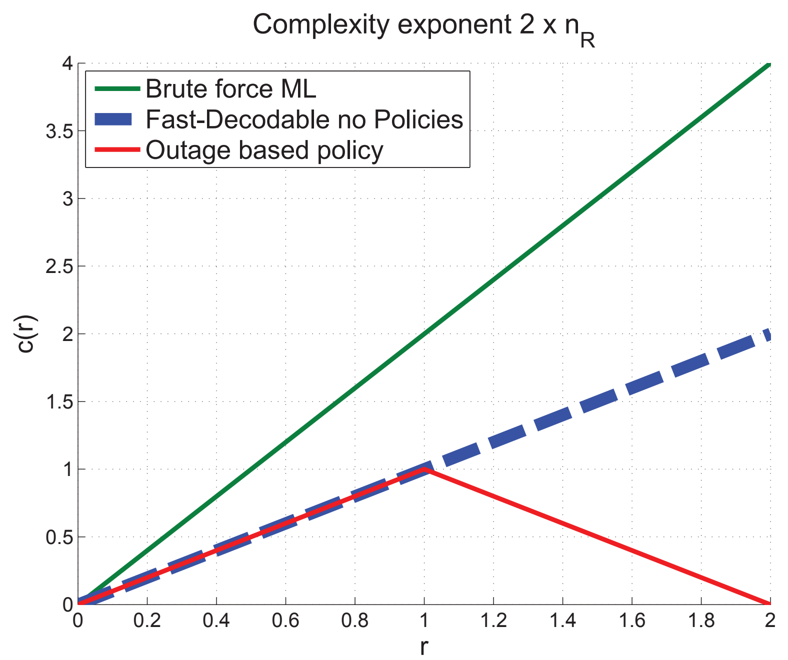
$$c(r) \leq \bar{c}(r) = \frac{T}{n_T} r(n_T - r), \quad r = 0, 1, \cdots, n_T$$

for all fading statistics, all full rate lattice designs, and all $\underbrace{\text{decoding order policies}}_{?}$.

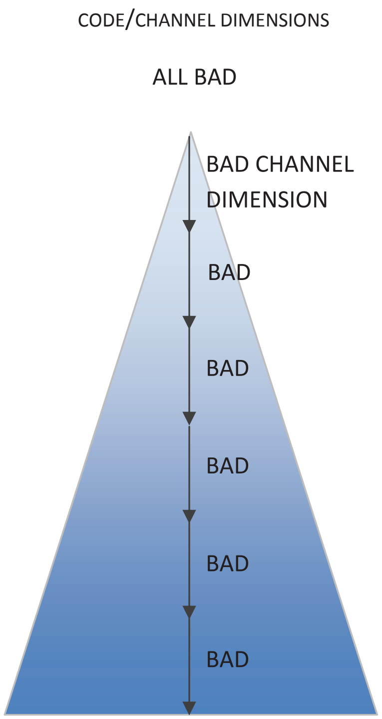
Example: $2 \times n_{\mathrm{R}}$ Quasi-static

EXAMPLE: ($2 \times n_{\mathrm{R}}$ channel ($n_{\mathrm{R}} \geq 2$))

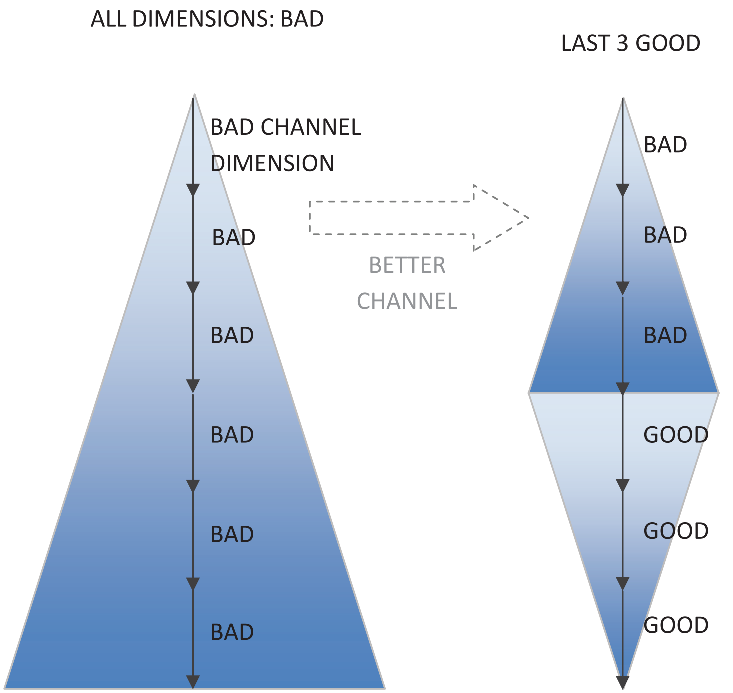
$$\bar{c}(r) = \begin{cases} r & r \leq 1, \\ 2 - r & r \geq 1. \end{cases}$$



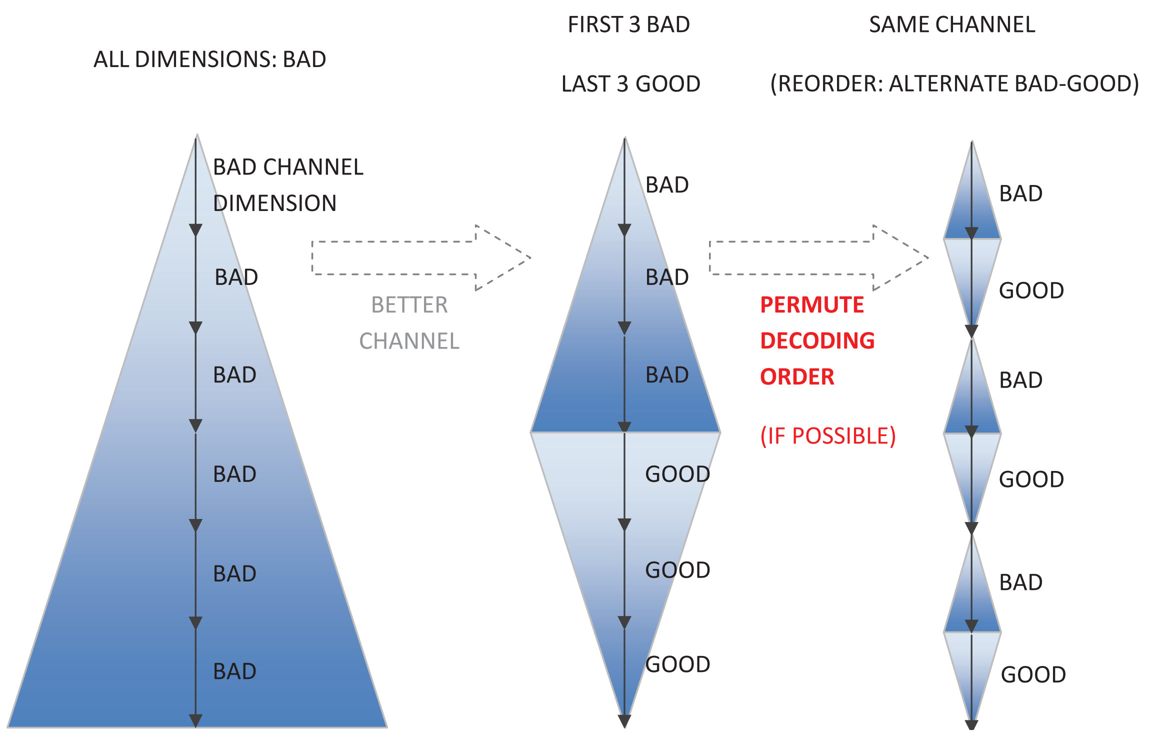
Decoding ordering and computational-halting policies



Decoding ordering and computational-halting policies$_1$



Decoding ordering and computational-halting policies$_2$



Tightness: DMT-opt quasi-static

Tightness of universal bound

Theorem: *(Quasi-static, Rayleigh, $n_R \geq n_T$) With probability 1 in the choice of the DMT optimal lattice design, the above is tight for all ordering policies.*

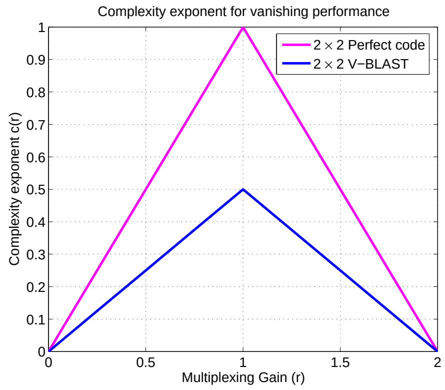
Theorem: *(Quasi-static, Rayleigh, $n_R \geq n_T$) The bound is tight for all layered designs, for several fixed orderings including the natural ordering.*

(Some hope remains for complexity reductions using dynamic policies)

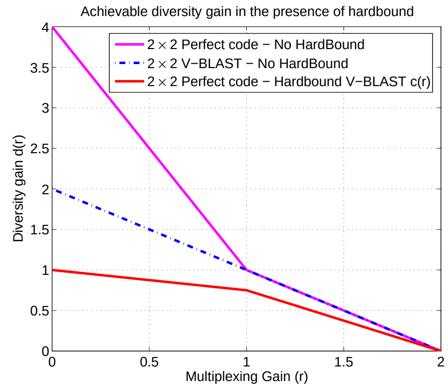
Complexity-constrained DMT

FIND MAX $d(r)$ GIVEN COMPLEXITY CONSTRAINT $N_{\max} \doteq \rho^{c_{\mathcal{D}}(r)}$ FLOPS

From uncoded to coded without increasing resources: a good idea?



(a) Complexity Hardbound



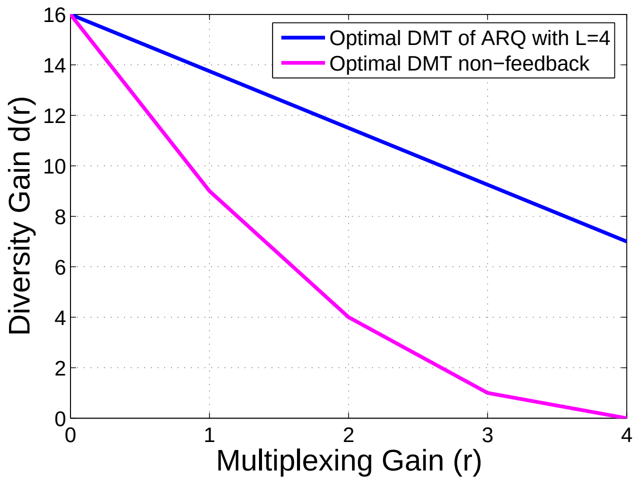
(b) Achievable DMT

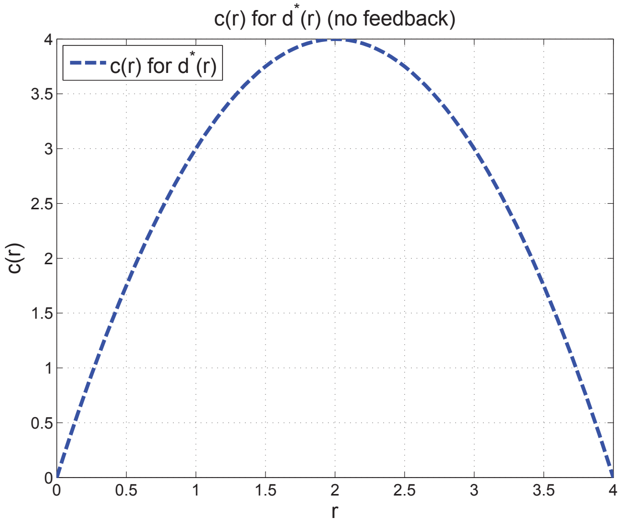
Performance-Complexity ramifications of feedback

Performance-Complexity ramifications of feedback

Two interesting questions:

- What is the feedback-aided complexity to achieve DMT $d^*(r)$?
- What is the complexity to achieve the feedback-aided DMT[1] $d^*(r/L)$?





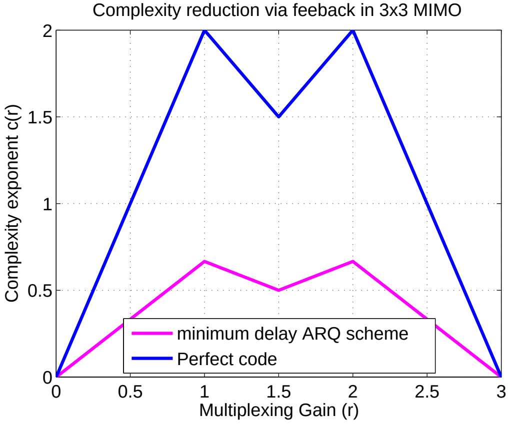
[1] El Gamal, Caire, Damen

Feedback-aided complexity for optimal DMT $d^*(r)$

Corollary: *(quasi-static iid Regular $n_R \geq n_T$, $LT = n_T$)*
Minimum $c(r)$ for $d^*(r)$, (minimized over all lattice designs, all L-round ARQ schemes, all halting and decoding order policies), bounded as (piecewise linear $r = 0, 1, \cdots, n_T$)

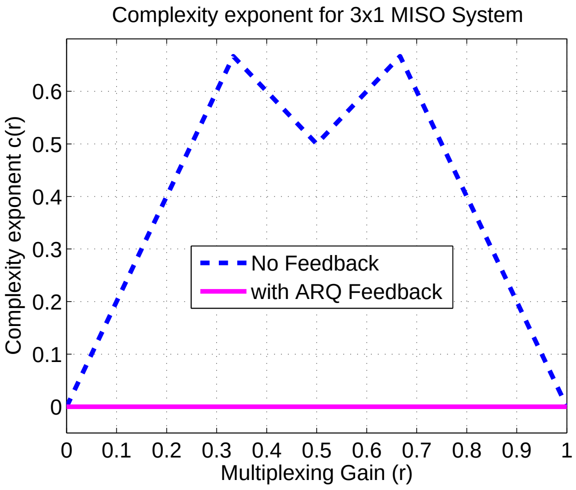
$$c(r) \leq \bar{c}_{red}(r) = \frac{1}{n_T} r(n_T - r).$$

- Compare to $c(r) = r(n_T - r)$
- Important role of "aggressive intermediate halting policies"



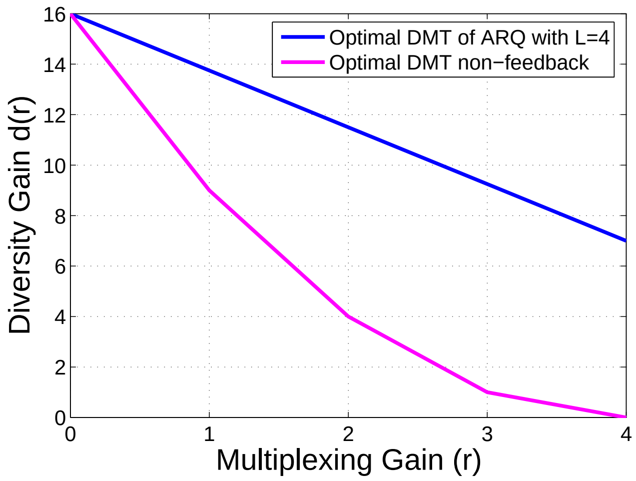
Feedback aided complexity for optimal DMT ($n_{\mathrm{R}} < n_{\mathrm{T}}$)

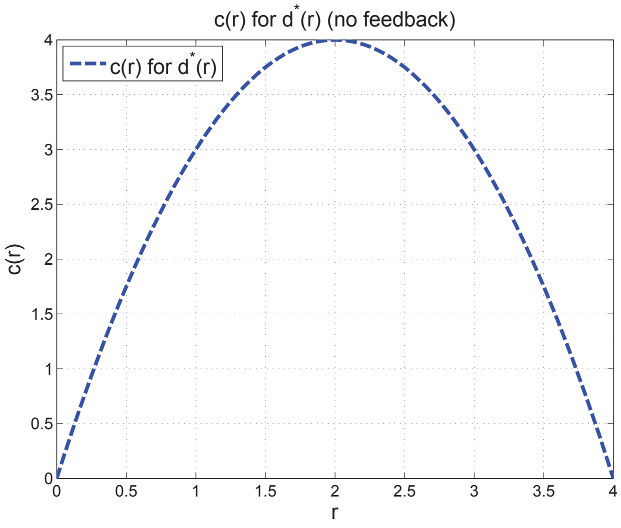
Corollary: $n_{\mathrm{T}} \times 1$ *MISO* $L = n_{\mathrm{T}}$, *then* $c(r) = 0$ *for* $d^*(r)$



SEEKING TO $c(r)$ NEEDED TO ACHIEVE $d^*(r/L)$

Recall:





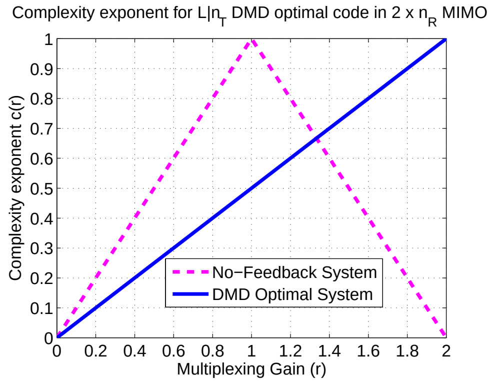
Complexity reduces with feedback despite increased $d^*(r/L)$

Theorem: *($L|n_T$, quasi-static, $n_R \geq n_T$)*
Minimum $c(r)$ to achieve optimal $d^*(r/L)$ is bounded as ((mult. of L))

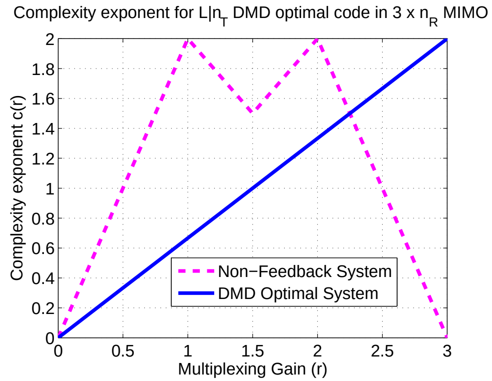
$$c(r) \leq \overline{c}_{dmd}(r) = \frac{r n_T}{L^2}\left(L - \frac{r}{n_T}\right).$$

Corollary: *The above with $L = n_T$ gives*

$$c(r) \leq \overline{c}_{DMD}(r) = \left(1 - \frac{1}{n_T}\right) r.$$



$2 \times n_R$

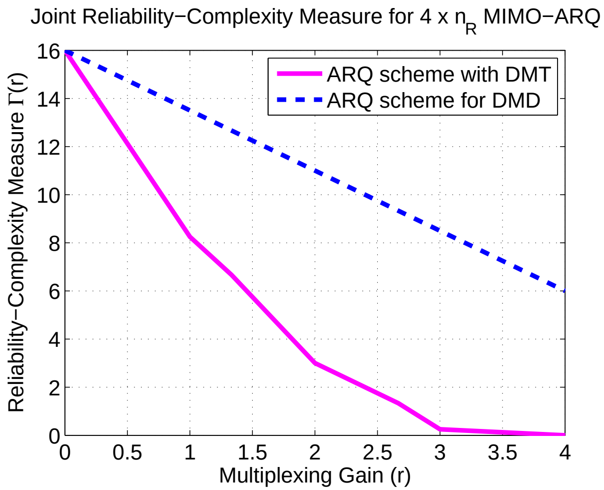


$3 \times n_R$

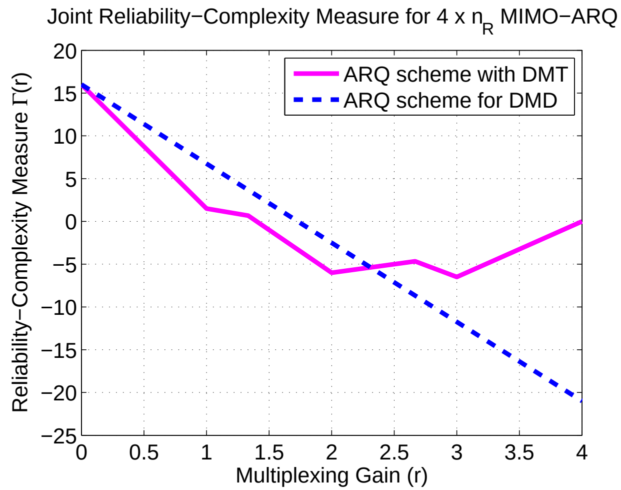
Have feedback: Go for basic DMT or feedback-aided DMT?

Joint performance-complexity measure

$$\Gamma(r) = d(r) - \gamma c(r)$$



$\gamma = 1$



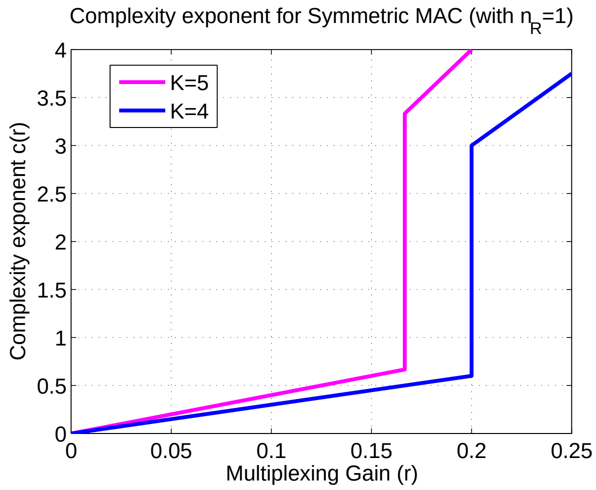
$\gamma = 10$

Joint reliability-complexity measure for DMT and DMD optimal ARQ schemes

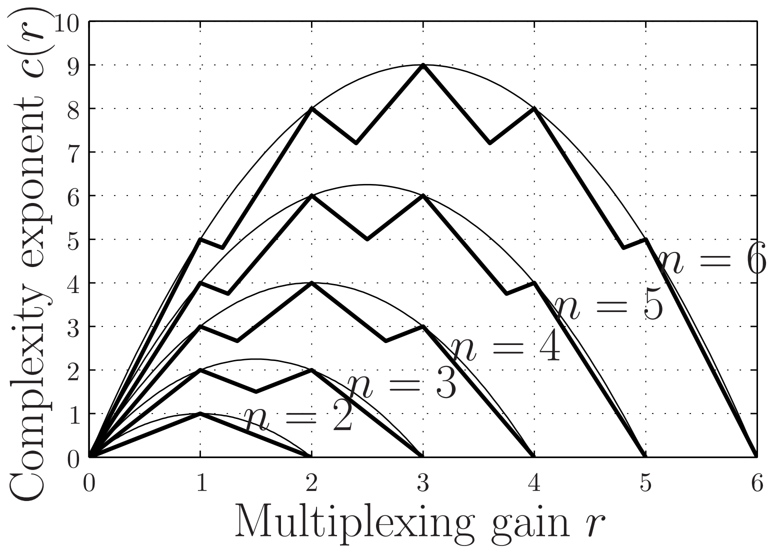
MIMO MAC

Corollary: *(K user MAC, $n_{\mathrm{T}} = 1$, $n_{\mathrm{R}} = 1$, r per user, Rayleigh, K odd)* *The minimum c(r) (over all lattice designs and halting and decoding order policies) to achieve the optimal MAC-DMT, is upper bounded as*

$$c(r) \leq \overline{c}_{mac}(r) = \begin{cases} (K-1)r & \textit{for } r \leq \frac{1}{K+1}, \\ (K-1)Kr & \textit{for } \frac{1}{K+1} < r \leq \frac{1}{K}. \end{cases}$$



Conclusion on search based algorithms



Conclusion for ML and (regularized) lattice based solutions

- Very considerable complexity for high performance
  - ⋆ Feedback helps
- No known way to drop below the upper bounds

Lattice reduction

Lets get some help from lattice reduction (LR)

But remember

LR Problematic in unavoidable scenarios (inner-outer code)

Lattice reduction

CHANGE REPRESENTATION OF LATTICE

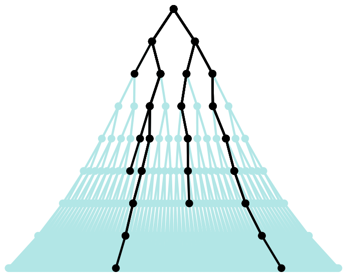
LR: Input $\boldsymbol{H}$, Output $\boldsymbol{T}$ (unimodular matrix of integers)

$$\underbrace{\boldsymbol{H}}_{\text{CHANNEL}} \cdot \underbrace{\boldsymbol{s}}_{\in \mathbb{Z}} = \underbrace{\boldsymbol{H}\boldsymbol{T}^{-1}}_{\substack{\text{better}\\ \text{channel}}} \cdot \underbrace{\boldsymbol{T}\boldsymbol{s}}_{\substack{\text{still}\\ \in \mathbb{Z}}},$$

Achieving optimal exponents - but unbounded gap

Theorem: *(Trans-IT Oct 2010) LR-aided reguralized linear decoding and LR-based halting, achieves $c(r) = 0, d(r) = d_{opt}(r)$, for all r, all codes, all MIMO scenarios and all fading statistics. (at most $O(n^2)$ flops per bit)*

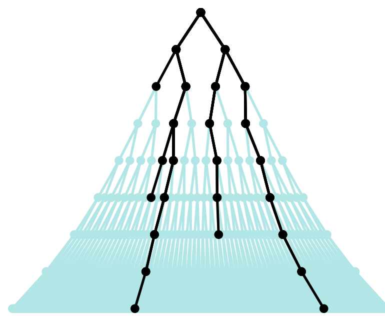
- First ever solution to achieve optimal $d^*(r)$ with subexponential complexity.



- BUT! Potentially <u>unbounded gap</u> to exact lattice decoding!

Achieving a vanishing gap at subexponential complexity

Theorem: *(Trans-IT subm. July 2011) LR-aided regularized lattice sphere decoding with LR- and outage-based halting policies, introduces a zero complexity exponent, and achieves a vanishing gap to the exact implementation of lattice decoding (all MIMO scenarios, all statistics, all codes).*



- First ever to achieve a vanishing gap to the exact solution of (regularized) lattice decoding, with subexponential computational complexity
- Again though - remember LR limitation!

Conclusions: Flops for Ergodicity

Can small chips (rather than CSIT) give us ergodicity?

- With LR - Yes: for a very broad setting, but not for near-ergodic rates
- With LR and 1 bit of feedback - Yes: for all r
- BUT: LR might not apply
- Without LR: Mostly NO - open problem - there might be hope!
  - ⋆ A little bit of feedback goes a long way
  - ⋆ High multiplexing gain most problematic

Other research contributions

Other contributions

- Cooperative wireless networks (Trans. IT 2009)+(subm. Trans-IT 2011)
- Two-way multi-directional communications
  - ⋆ patents (pending) - publications - funding - award
- Cross layer optimization - queue/channel (Trans. IT 2009)
  - ⋆ Towards a consummated union: finite delay results
- Feedback (Trans. IT 2009)
- Connectivity in networks with bounding constraints (publications)

Other contributions$_1$

- Soft-biometrics / surveillance networks / computer vision (publications)[2]
  - ⋆ Will spend some time at well known biometrics lab in US
- Interference (preliminary work)
  - ⋆ Interference alignment and diversity (potential submission ISIT-2012)
  - ⋆ Stale CSIT (potential submission ISIT-2012)
  - ⋆ Uplink-downlink DOF (ITA-2011, ISIT-2011)
  - ⋆ Finite SNR IA. LR-aided IA.

[2] "Search pruning video surveillance systems: Efficiency-reliability tradeoff," 1st IEEE Workshop on Information Theory in Computer Vision and Pattern Recognition, Nov. 2011.

Other than that

- Successful funding efforts (e.g. ANR Blanc International)(also on surveillance)
- Relatively close to industry

- Tutorials + awards (for last two results + two-way)
- Efforts to recruit talented students

Thank you

Thank you very much for coming