Construction of Length and Rate Adaptive MET QC-LDPC Codes by Cyclic Group Decomposition

Vasiliy Usatyuk, Egorov Sergey
South-West State University,
Kursk, L@Lcrypto.com, sie58@mail.ru

German Svistunov,
Omsk State Technical University,
Omsk, g.v.svistunov@gmail.com

Batumi, Georgia
September 13 - 16, 2019

$$H = \begin{array}{c} \begin{array}{cccccc} v_1 & v_2 & v_3 & v_4 & v_5 & v_6 \end{array} \\ \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ \hline 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \end{array} \begin{array}{c} c_1 \\ c_2 \\ c_3 \\ c_4 \end{array}$$

Quasi-Cyclic LDPC(QC-LDPC codes) - LDPC-codes with parity-check matrix defined by structured block submatrix – Circulant Permutation matrix.

$$H_{QC} = \begin{bmatrix} I^0 & I^1 & I^1 \\ I^0 & I^{-1} & I^0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 \\ 0 & -1 & 0 \end{bmatrix}$$

$$Circulant\ \ Permutation\, Matrix\,(CPM)\ of\ \ size\ \ 2\times 2\colon\ I^0 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I^1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, I^{-1} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

R. M. Tanner, D. Sridhara, T. Fuja, ”A class of group structured LDPC codes”, Proc. ICSTA 2001, Ambleside, England, 2001

Length adaptation of QC-LDPC Codes

8x16 protograph base matrix

$$H_{proto} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Circulant size 42

$$H = \begin{pmatrix} 33 & 0 & 15 & 0 & 8 & 0 & 28 & 0 & 26 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 14 & 0 & 25 & 0 & 11 & 0 & 0 & 9 & 18 & 33 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 8 & 0 & 11 & 0 & 29 & 0 & 30 & 0 & 20 & 18 & 0 & 0 & 0 & 0 & 0 \\ 0 & 31 & 0 & 39 & 0 & 22 & 31 & 0 & 0 & 0 & 37 & 11 & 0 & 0 & 0 & 0 \\ 33 & 0 & 9 & 0 & 5 & 0 & 24 & 0 & 25 & 0 & 0 & 28 & 23 & 0 & 0 & 0 \\ 20 & 0 & 30 & 0 & 0 & 20 & 0 & 12 & 0 & 30 & 0 & 22 & 0 & 12 & 0 & 0 \\ 2 & 20 & 0 & 11 & 0 & 31 & 0 & 7 & 0 & 0 & 36 & 0 & 0 & 17 & 6 & 0 \\ 0 & 7 & 0 & 32 & 24 & 0 & 39 & 0 & 30 & 0 & 0 & 0 & 26 & 0 & 38 & 28 \end{pmatrix}$$

Code length N=16*42=672

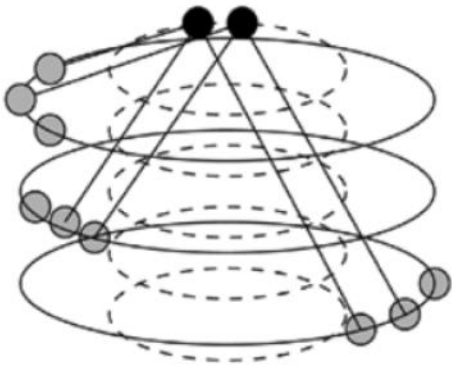
Circulant size 9

$$H = \begin{pmatrix} 2 & -1 & 6 & -1 & 7 & -1 & 7 & -1 & 8 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 7 & -1 & 6 & -1 & 1 & -1 & -1 & 3 & 6 & 3 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & 2 & -1 & 8 & -1 & 2 & -1 & 8 & -1 & 1 & 8 & -1 & -1 & -1 & -1 & -1 \\ -1 & 5 & -1 & 6 & -1 & 2 & 4 & -1 & -1 & -1 & 0 & 1 & -1 & -1 & -1 & -1 \\ 4 & -1 & 6 & -1 & 3 & -1 & 1 & -1 & 5 & -1 & -1 & 5 & 4 & -1 & -1 & -1 \\ 7 & -1 & 4 & -1 & -1 & 2 & -1 & 0 & -1 & 6 & -1 & 5 & -1 & 4 & -1 & -1 \\ 8 & 1 & -1 & 4 & -1 & 4 & -1 & 5 & -1 & -1 & 8 & -1 & -1 & 2 & 7 & -1 \\ -1 & 1 & -1 & 3 & 4 & -1 & 5 & -1 & 3 & -1 & -1 & -1 & 4 & -1 & 4 & 3 \end{pmatrix}$$

Code length N=16*9=144

$$E(H_{current}) = E(H_{upper}) \bmod z_{current}$$

$z_{current}$ – *circulant size which you want to get*



For m to n protograph

- code bit nodes (n rings of L_K each)
- constraint nodes(m rings of L_K each)

We can represent 8 cyclic groups of 5G as:

$$L_k = a \times 2^i, \quad i = \{0,1,\ldots,7\}$$

$$a = \{2,3,5,7,9,11,13,15\}$$

$$L_{256} = L_2 \times L_2 \times \cdots \times L_2$$

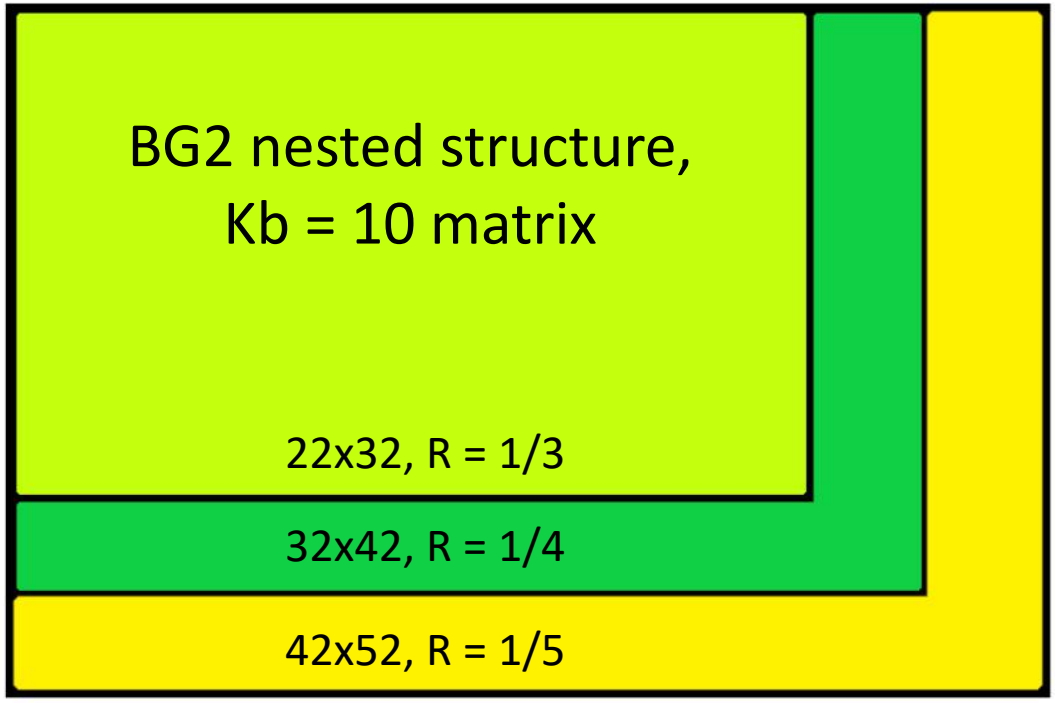
$$L_{384} = L_3 \times L_2 \times \cdots \times L_2$$

$$L_{320} = L_5 \times L_2 \times \cdots \times L_2$$

$$\ldots$$

$$L_{240} = L_{15} \times L_2 \times \cdots \times L_2$$

• 3GPP TS38.212V15.4.0:NR;Multiplexing and channel coding (Rel. 15)
• Tanner R.M., Sridhara D., Sridharan A., Fuja T.E., and Costello D.J., “LDPC block and convolutional codes based on circulant matrices”, IEEE Trans. Inf. Theory, vol. 50, no. 12, pp. 2966-2984, Dec. 2004.
• Seho Myung; Kyeongcheol Yang, "Extension of quasi-cyclic LDPC codes by lifting," in Information Theory, 2005. ISIT 2005. Proceedings. International Symposium on, vol., no., pp.2305-2309, 4-9 Sept. 2005
• Seho Myung; Kyeongcheol Yang; Youngkyun Kim, "Lifting methods for quasi-cyclic LDPC codes," in Communications Letters, IEEE , vol.10, no.6, pp.489-491



3GPP TS38.212V15.4.0:NR;Multiplexing and channel coding (Rel. 15)

Algorithm 1 Codes Sieving Method For Construction of Length Adaptive MET QC-LDPC Codes

Require: $M(H)-$ mother matrix, L_0-maximal lifting value, $K-$set of circulant sizes for code distance sieving, e.g. $\{4, 8, 16, 32\}$, $Card_c$-number of lifted codes for sieving.

1: $E_{sieve}(\mathbf{H}) = \emptyset$
2: **for** $i = 0; i < Card_c; i = i + 1$ **do**
3: $E(\mathbf{H})_i = SALift(M(\mathbf{H}), L_0)$
4: $E(H_K) = f_{modular}(E(\mathbf{H})_i, K)$
5: $dmin_{K,i} = NG(E(H_K))$
6: **end for**

return $E_{sieve}(\mathbf{H}) = \max_{dmin_{K,i}} E(H_K)$

where $E(\mathbf{H})_i$ - is the set of codes constructed by simulation annealing lifting $E(\mathbf{H}) = \max_{girth,EMD} M(\mathbf{H})$, $E(H_K)$ - is the length adapted codes obtained by using modular lifting for circulant size from set $\{K\}$, $dmin_{K,i}-$ is a code distance estimated by Alg. 2.

*Simulated Annealing Method for QC lifting

**8 cyclic group decomposition Modular length adaption

Number Geometry Probabilistically

Code Distance (Hamming Distance) Estimation

Source code published: Construction of Length and Rate Adaptive MET QC-LDPC https://github.com/Lcrypto/

*V. Usatyuk and I. Vorobyev, "Simulated Annealing Method for Construction of High-Girth QC-LDPC Codes," *2018 41st International Conference on Telecommunications and Signal Processing (TSP)*, Athens, 2018, pp. 1-5.
https://github.com/Lcrypto/Simulated-annealing-lifting-QC-LDPC

**I. Vorobyev, N. Polyanskii, G. Svistunov, S. Egorov and V. Usatyuk, "Generalization of Floor Lifting for QC-LDPC Codes: Theoretical Properties and Applications," *2018 IEEE East-West Design & Test Symposium (EWDTS)*, Kazan, 2018, pp. 1-6.; S. Egorov, V. Usatyuk Generealization of floor lifting Novi Sad. EWDTS 2017

$Let\ H^{(n-k)\times n} \in F_q, which\ define\ map\ F_q^n \to F_q^{n-k}, x \to Hx,$
$\ker(H) = \{c \in F_q^n \,|Hc = 0\}$, where $H-$parity$-$check matrix.

Code (Hamming distance) distance problem for a linear block code.
Find a vector c with a minimum number of non-zero coordinates,

$$c \in \ker(H) \setminus \{0\}^k.$$

Minimum distance of a linear code is not aproximable to within any constant factor in random polynomial time (RP), unless nondeterministic polynomial time (NP) equals RP.
Dumer I, Micciancio D., Sudan M. Hardness of Approximating the Minimum Distance of a Linear Code ISIT 2003

$$Hc = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$GF(2)$

$$\bar{c}_0^{\,T} = [0 \quad 0 \quad 0 \quad 0 \quad 0] \qquad w(\bar{c}_0^{\,T}) = \left\| \bar{c}_0^{\,T} \right\| = 0$$

$$\bar{c}_1^{\,T} = [0 \quad 0 \quad 0 \quad 1 \quad 1] \qquad w(\bar{c}_1^{\,T}) = \left\| \bar{c}_1^{\,T} \right\| = 2$$

$$\bar{c}_2^{\,T} = [0 \quad 1 \quad 1 \quad 0 \quad 1] \qquad w(\bar{c}_2^{\,T}) = \left\| \bar{c}_2^{\,T} \right\| = 3$$

$$\bar{c}_3^{\,T} = [0 \quad 1 \quad 1 \quad 1 \quad 0] \qquad w(\bar{c}_3^{\,T}) = \left\| \bar{c}_3^{\,T} \right\| = 3$$

weight 0 vector

$$W(c) = [1, 0, 1, 2, 0, 0]$$

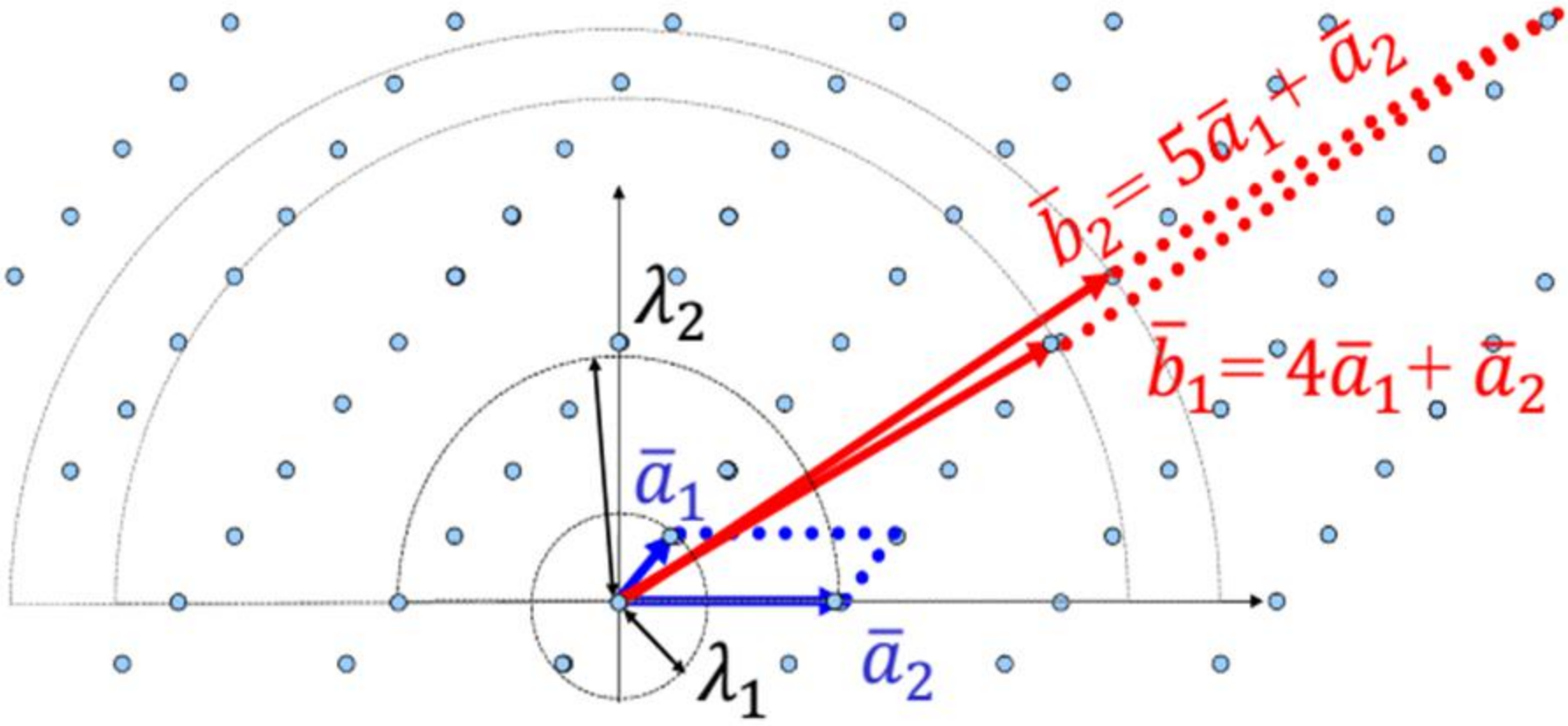
$$f(z) = 1 + z^2 + 2z^3$$

weight 2 vector

weight 3 vectors

$$d_{\min} = 2;$$

$Lattice\ L$ – discrete Abelian subgroup under Euclidean space, $B = \{b_1, \dots, b_m\} \subset R^m$
$L(\boldsymbol{B}) = \{\sum_{i=1}^{n} x_i b_i : x_1, \dots, x_n \in Z^n\}$,



Same Lattice $L(\boldsymbol{B_1}) = L(\boldsymbol{B_2})$, $\boldsymbol{B_1} = \{(\mathbf{1},\mathbf{1}),(\mathbf{4},\mathbf{0})\}$, $\boldsymbol{B_2} = \{(\mathbf{8},\mathbf{4}),(\mathbf{9},\mathbf{5})\}$

$$\boldsymbol{det|B_1| = det|B_2|}$$

Algorithm 2 Number Geometry based probabilistic code distance estimation method

Require: G—Code generator matrix, $Type$—type of searching area, d_{min}^{upper}—upper bound on code distance, Num—number of random permutation of lattice basis, q—code alphabet $q \in 2,3$, β—block size in BKZ-basis reduction method, δ—precious of length reduction.

1: Embedded code to lattice

$$B_c^T = \begin{pmatrix} N \cdot G & I_k \\ N \cdot q \cdot I_n & 0^{n \times k} \end{pmatrix}$$

2: B'=$SBP_{\Delta}(m)$ using BKZ(β, δ)

3: Generate Num permutation of basis B'

4: **for** $basis = 0; basis <= Num; Num = Num + 1$ **do**

5: $QR\left(perm_{basis}\left(B\right)\right)$

6: Exp_{basis}^{Type} and Var_{basis}^{Type}

7: **end for**

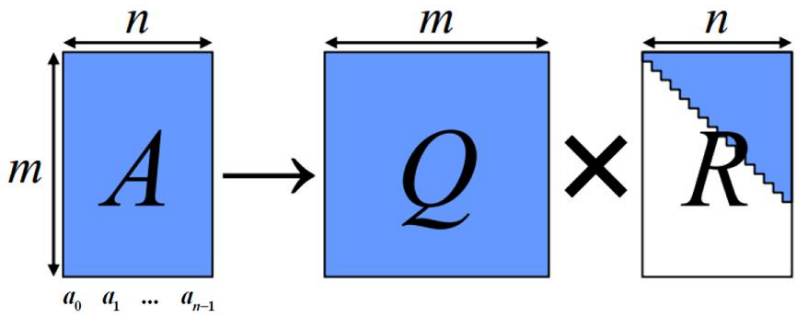
8: $B^* = min\left(Exp_{basis}, Var_{basis}\right)$

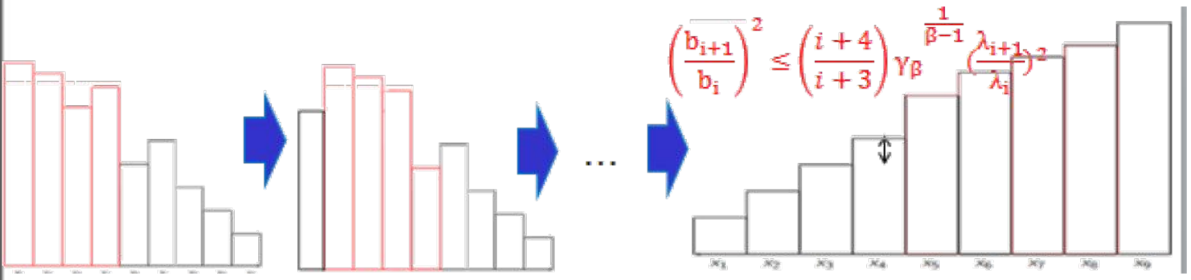
9: $c = SVP_{\Delta}L\left(B^*\right)$ with radius $R \leq d_{min}^{upper}$, area $Type$

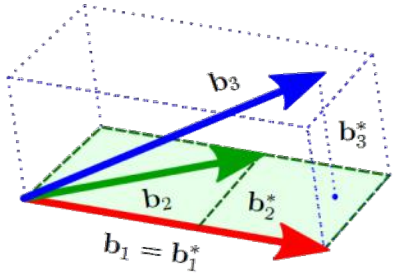
return d_{min} number of non-zero position in c

Block Korkin-Zolotarev method (BKZ(β)):

- LLL=BKZ(β=2) prereduction ;
- $i = 0; i = (i(mod(m-1)) + 1.$
  $b_i' : \left\|b_i'\right\| = \lambda_1(L(\pi_i(b_i), ..., \pi_i(b_{\min(i+\beta-1,m)})))$
  IF $\delta^2 \|b_i^*\|^2 > \left\|b_i'\right\|^2$:
  - $B = \{b_1, ..., b_m\} := \{b_1, ..., b_{i-1}, b_i', b_{i+1}, ..., b_m\}$;
- LLL=BKZ(β=2) postreduction ;
- return basis, if it don't change after m-1 itteration.







Proposed method for e.x. allow to improve code distance, e.g. for family a=2, for CPM size 8 from **20** (5G) to **23(Our)**, for CPM size 32 from **31** (5G) to **44 (Our)**.

Probabilistically shortest vector problem

Shortest vector problem for search vector in sphere S could be relaxed by choice area P, $u = \dim(P)$, where $S -$ sphere radius R.

Lattice of dimension m, $x \in L \bigcap S \bigcap \mathrm{P}$ u << m.

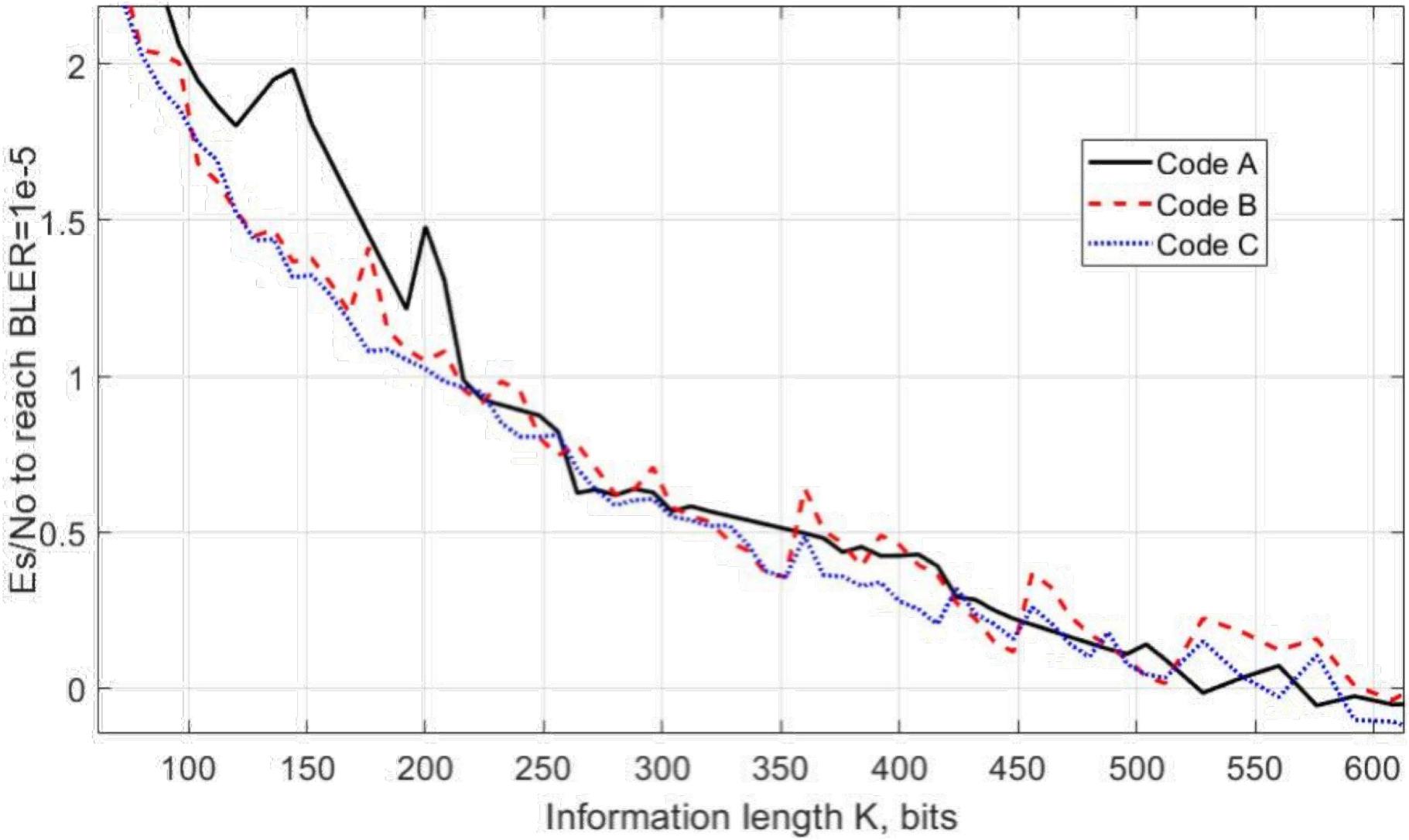
$$\left\| \sum_{i=1}^{m} x_i b_i^* \right\|_2^2 \le \mathrm{R}^2, \quad \underbrace{\begin{array}{c} -1/2 \le x_i < 1/2, i \le m-(u+1) \\ -1 \le x_i < 1, m-u \le i \le m-1 \\ 1/2 \le x_i < 3/2, i = m \end{array}}_{\mathrm{P}}$$

Shortest vector Exp, Var of found x in area P:

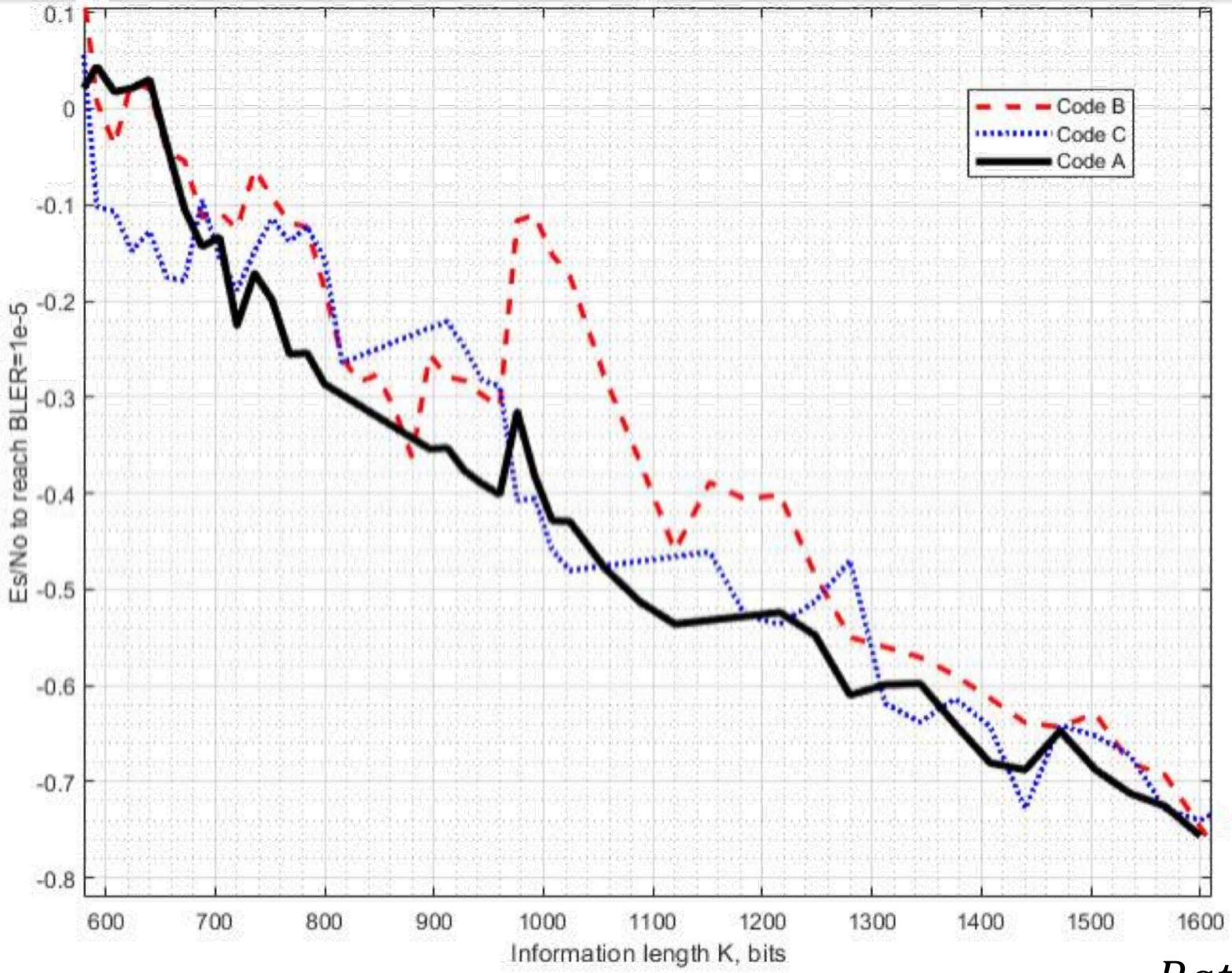
$$Exp = \sum_{i=1}^{n} \left(\frac{t_i^2}{4} + \frac{t_i}{4} + \frac{1}{12} \right) \left\| b_i^{\perp} \right\|^2, \; Var = \sum_{i=1}^{n} \left(\frac{t_i^2}{48} + \frac{t_i}{48} + \frac{1}{180} \right) \left\| b_i^{\perp} \right\|^4.$$

Schnorr C.P. Lattice Reduction by Random Sampling and Birthday Methods. STACS 2003.Lecture Notes in Computer Science

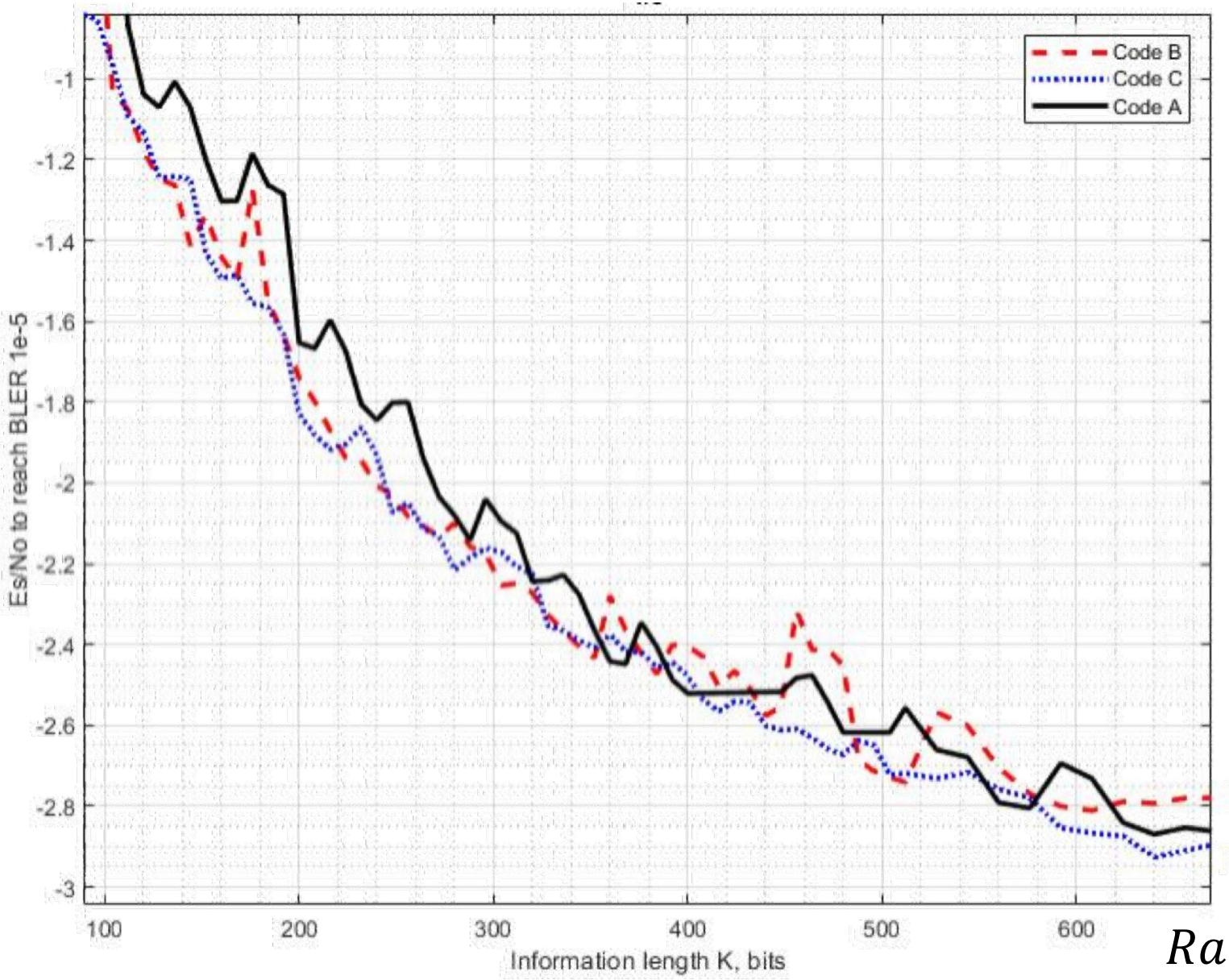
Yoshinori A., P. Q. Nguyen Random Sampling Revisited: Lattice Enumeration with Discrete Pruning EuroCrypt 2017


Es/No to reach BLER=1e-5
Information length K, bits
Code A
Code B
Code C
0
0.5
1
1.5
2
100
150
200
250
300
350
400
450
500
550
600

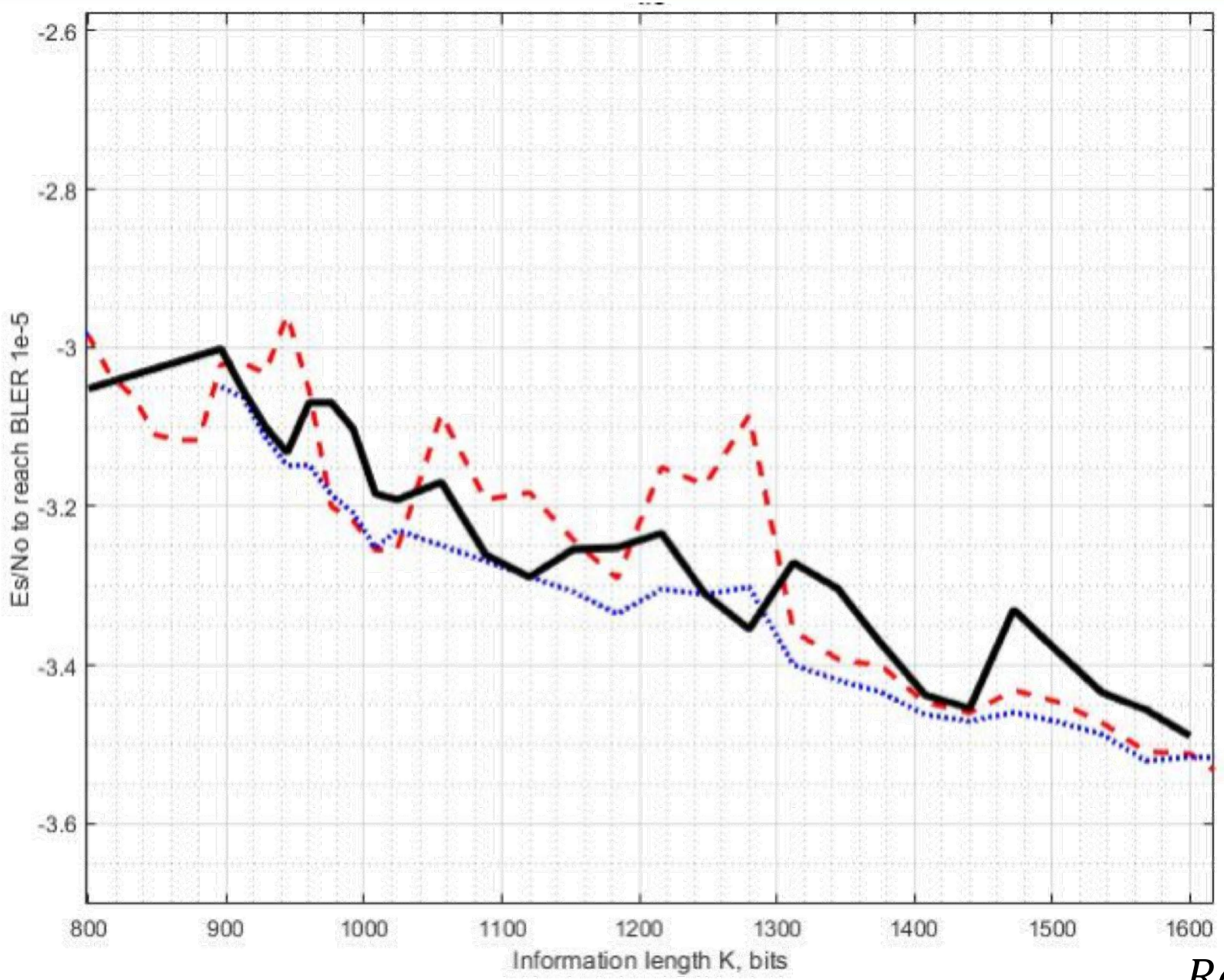
$Rate\ 1/3$


Code B
Code C
Code A
Es/No to reach BLER=1e-5
Information length K, bits
0.1
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
600
700
800
900
1000
1100
1200
1300
1400
1500
1600

Rate 1/3


Code B
Code C
Code A
Es/No to reach BLER 1e-5
-1
-1.2
-1.4
-1.6
-1.8
-2
-2.2
-2.4
-2.6
-2.8
-3
100
200
300
400
500
600
Information length K, bits

Rate 1/5


-2.6
-2.8
-3
-3.2
-3.4
-3.6
Es/No to reach BLER 1e-5
800
900
1000
1100
1200
1300
1400
1500
1600
Information length K, bits

Rate 1/5

Thank You!

Questions?

Number Geometry Probabilistically Code Distance Estimation
for improving properties of code properties (under short and medium length)

Knapsack problem:

$$\sum_{i \in I} a_i = s$$

$$A = \{a_1, a_2, \ldots, a_n\}, I \subseteq \{1,2,\ldots,n\}, a, s \in Z^+$$

Equivalent to Knapsack problem Shortest vector problem

$$B_k = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & 0 & -a_2 \\ 0 & 0 & 1 & \cdots & 0 & -a_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -a_n \\ 0 & 0 & 0 & \cdots & 0 & s \end{pmatrix}$$

J. C. Lagarias, A. M. Odlyzko. Solving low-density subset sum problems. Journal of the Association for Computing Machinery 32 (1985) 229–246.

$$B_k = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 & -a_1 \\ 0 & 1 & 0 & \cdots & 0 & -a_2 \\ 0 & 0 & 1 & \cdots & 0 & -a_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -a_n \\ 0 & 0 & 0 & \cdots & 0 & s \end{pmatrix}$$

If exist solution of Knapsack problem

$$x_1 a_1 + x_2 a_2 + \ldots + x_n a_n = s$$

$$x_1, x_2, \ldots, x_n \in \{0,1\}$$

then exit short vector in Lattice

$$\bar{v} = x_1 M_1 + x_2 M_2 + \ldots + x_n M_n + M_{n+1} = [x_1, x_2, \ldots, x_n, 0]$$

Kannan's embedding technique

Ravi Kannan Minkowski's convex body theorem and integer programming. Mathematics of operations research, 12(3):415–440, 1987.

Lattice basis $B_c = \begin{pmatrix} G & qI_n \\ I_k & 0 \end{pmatrix}$ B_c *size* $(n+k)\times(n+k)$

G *size* $k \times n$

Kannan's embedding technique

***G*-generator matrix of linear block code.**

$q \in \{2,3\}-$ **Code alphabet,** k **- information length,** $n-$ **code length**

Block Korkin-Zolotarev method (BKZ(β)):

1. LLL=BKZ(β=2) prereduction ;
2. $i = 0; i = (i(mod(m-1)) + 1.$
   $b_i' : \left\|b_i'\right\| = \lambda_1(L(\pi_i(b_i), ..., \pi_i(b_{\min(i+\beta-1,m)})))$
   IF $\delta^2 \|b_i^*\|^2 > \left\|b_i'\right\|^2$:
   - $B = \{b_1, ..., b_m\} := \{b_1, ..., b_{i-1}, b_i', b_{i+1}, ..., b_m\}$;
3. LLL=BKZ(β=2) postreduction ;
4. return basis, if it don't change after m-1 itteration.

After lattice reduction And search shortest vector

Ravi Kannan Minkowski's convex body theorem and integer programming. Mathematics of operations research, 12(3):415–440, 1987.

Example of Code distance estimation for Goley code (11,6), with generator matrix:

$$G = \begin{pmatrix} 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 & 2 & 0 & 1 \end{pmatrix}$$

$$q = 3, GF(3)$$

$$B_c = \begin{pmatrix} N \cdot G' & N \cdot qI_n \\ I_k & 0 \end{pmatrix}$$

After Lattice basis reduction using Block Korkin-Zolotarev method with block size $\beta = 2$, scale coefficient N=6

$$B_c^T = \begin{pmatrix} 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 12 & 12 & 6 & 12 & 0 & 6 & 0 & 0 & 0 & 0 & 0 & 1 \\ 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

After basis reduction get basis of lattice:

$$B_c = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -3 & 0 \\
6 & 6 & 6 & 0 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & -1 & 0 & 1 & 0 & 1 & 0 \\
6 & 0 & 0 & -6 & 0 & -6 & 0 & 0 & -6 & 6 & 0 & -1 & 1 & 1 & -1 & 1 & 0 \\
0 & 6 & 6 & 6 & 0 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & -1 & 0 & 1 & 0 & 1 \\
6 & 0 & 0 & 0 & 6 & 0 & 0 & 6 & 0 & 6 & 6 & -1 & 1 & 1 & 1 & 1 & 1 \\
6 & 6 & 0 & -6 & -6 & 0 & 0 & -6 & 0 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & 0 \\
0 & -6 & 0 & 6 & 0 & 6 & -6 & 0 & 0 & 0 & 6 & 0 & 1 & -1 & 1 & 0 & 1 \\
0 & -6 & 0 & 0 & 0 & 0 & 0 & 6 & 6 & -6 & 6 & 0 & 1 & -1 & -1 & -1 & 1 \\
-6 & 0 & -6 & -6 & -6 & 0 & 0 & 0 & -6 & 0 & 0 & 1 & -1 & 0 & -1 & 0 & 0 \\
0 & 0 & -6 & 0 & -6 & -6 & -6 & 0 & 0 & 0 & -6 & 0 & 0 & 1 & -1 & 0 & -1 \\
0 & 0 & 0 & -6 & 0 & 0 & -6 & 0 & -6 & -6 & -6 & 0 & 0 & 0 & 1 & -1 & -1 \\
0 & 0 & -6 & 0 & 0 & -6 & 0 & -6 & -6 & -6 & 0 & 0 & 0 & 1 & -1 & -1 & 0
\end{pmatrix}$$

After drop 6 rows and last 6 columns,rescale basis. Get short vectors：

$$B_c = \begin{pmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\
1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\
1 & 1 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 1 \\
-1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0
\end{pmatrix}$$

$$B_c = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & -1 & 0 & 0 & -1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & -1 & -1 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 1 & -1 & 0 & 0 & 0 & 1 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & -1 & 1 \\ -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 \\ 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1 & -1 & 0 \end{pmatrix}$$

Using Kannan-Finke-Post method enumerate basis to be sure that short basis contain shortest vector. After full search get $v = (0,0,-1,0,0,-1,0,-1,-1,-1,0)$

Which equivalent to $(0,0,2,0,0,2,0,2,2,2,0)$ to codeword of weight 5.

$$d_{\min} = 5$$