



# 새로운 Golomb Ruler를 이용하여 생성된 QC-LDPC 코드 성능분석

김대경, 김인선, 조현우, 송홍엽  
연세대학교

제32회 통신정보 합동학술대회(JCCI 2022)

2022.04.28



# 목차

- 서론
  - 연구 동기
  - QC-LDPC 부호
  - Golomb Ruler
- 본론
  - 새로운 Golomb Ruler 생성법
  - 시뮬레이션 방식
  - 시뮬레이션 결과
- 결론
- 참고문헌



# 연구 동기

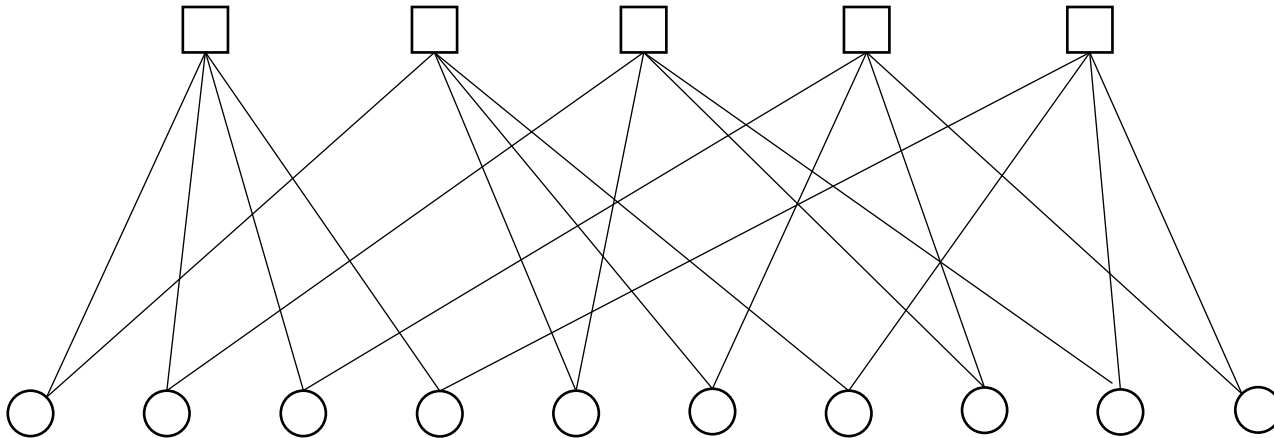
- Golomb Ruler를 이용해 girth 8 이상의 QC-LDPC 부호를 생성하는 내용의 논문[3]이 게재 예정임
- 이렇게 생성된 QC-LDPC 부호는 Golomb ruler의 마지막 mark를 증가 시킴에 따라 성능이 좋아지는 경향성을 보이다 어느 순간 성능이 나빠지는 경향성을 보임
- 해당 논문에서는 그 원인을 밝히지 못했고 추후 연구과제로 제시함
- 본 논문에서는 부호의 성능이 전환되는 부분을 집중적으로 살펴보고 실험한 결과에 대한 요약 정리



# LDPC 부호

- Low density parity check (LDPC) 부호
  - Parity check 행렬에서 '0'이 아닌 값을 갖는 원소가 sparse하게 존재하는 부호
  - Parity check 행렬을 bi-partite 그래프로 표현 (Tanner graph)
  - 반복 복호(iterative decoding) 방식을 사용
  - LDPC 부호의 parity check 행렬 → 랜덤 구조 또는 대수적 구조로 설계 가능

$$H = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$



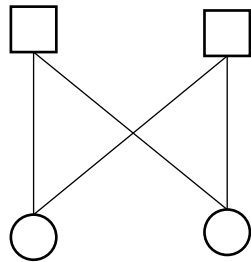


# LDPC 부호의 Cycle

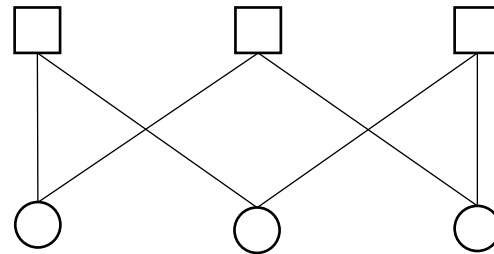
- Cycle

- Tanner 그래프의 한 노드에서 시작해서 다른 노드들을 한번씩만 지나서 원래 노드에 도달하는 경로
- Cycle의 길이 : cycle이 포함하는 edge의 수
- 길이가  $n$ 인 cycle을  $n$ -cycle이라 부름

- 4-cycle, 6-cycle 예시



4-cycle



6-cycle



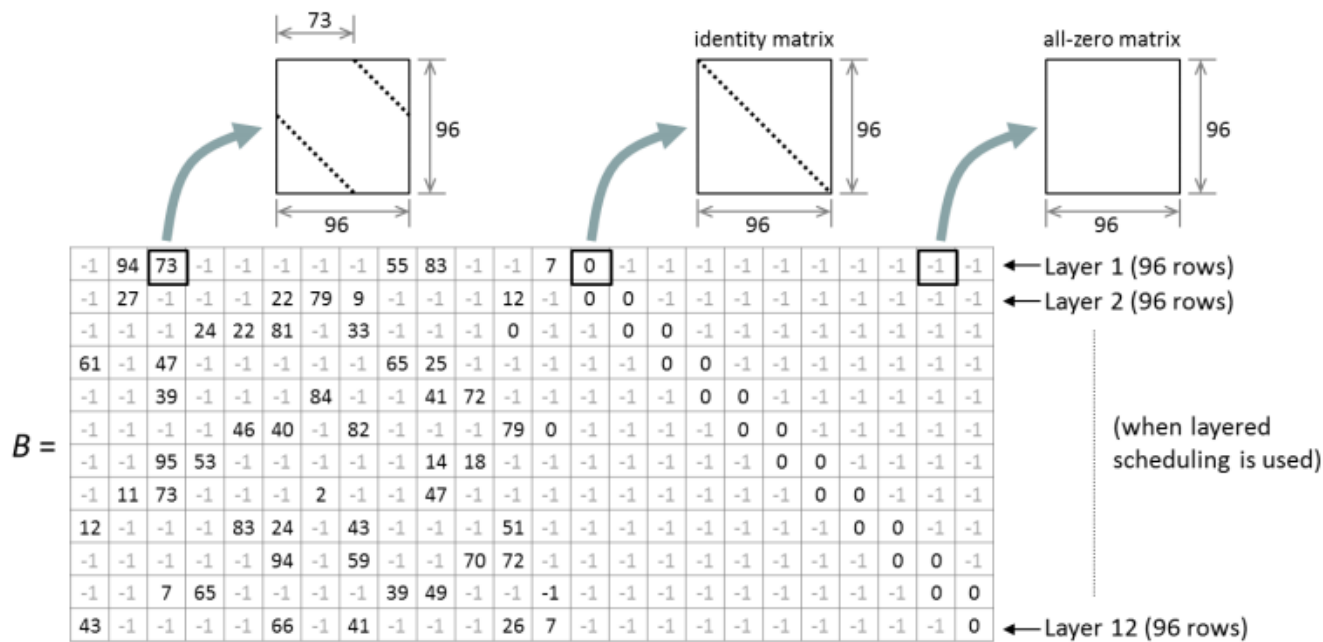
# Girth를 고려해야하는 이유

- Girth
  - Tanner 그래프에 존재하는 cycle들에 대해 최소 cycle 길이
- Sum-product 복호 결과에 성능 영향
  - girth가 무한대라면 sum-product 복호도 optimal ML 복호가 된다
  - girth가 작아질수록 sum-product 복호는 optimal ML 복호로부터 멀어진다



# Exponent 행렬을 이용한 QC-LDPC 부호 생성법

- Exponent 행렬을 이용한 QC-LDPC 부호 생성법
  - Exponent 행렬의 각 원소마다 그 값만큼  $P \times P$  항등 행렬  $I_p$ 를 순환 이동한 행렬을 대입하여 QC-LDPC 부호의 Parity-Check 행렬 생성 가능
- 예시
  - WIMAX LDPC 코드,  $N = 2304, P = 96$





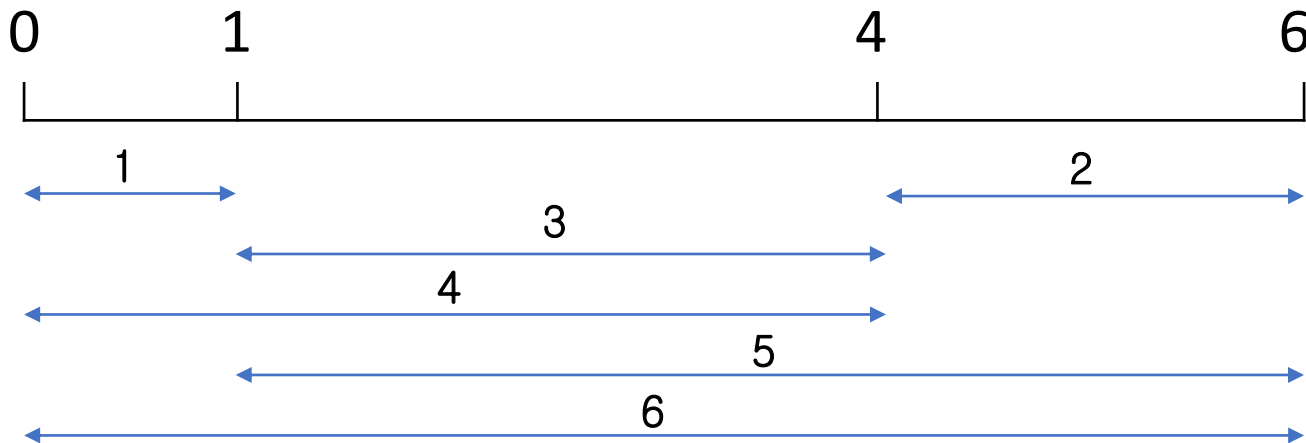
# Golomb Ruler

- Golomb Ruler

- $g_1 = 0 < g_2 < \dots < g_n$  를 만족하는  $n$  개의 정수  $g_1, g_2, \dots, g_n$  의 집합으로, 모든  $i < j$  에 대해서  $g_j - g_i$  가 모두 다르다는 성질로 정의
- $n$ -mark Golomb Ruler : mark의 개수가  $n$ 개인 Golomb Ruler

- 예시

- $\{g_1, g_2, g_3, g_4\} = \{0, 1, 4, 6\}$  ( 4-mark Golomb Ruler )



marker 2개 사이의 간격이 모두 다름





# Golomb Ruler를 이용한 Exponent 행렬 생성

- 최적 Golomb Ruler
  - Golomb Ruler의 길이  $L = g_n - g_1$
  - 최적  $n$ -mark Golomb Ruler는 최소 길이를 갖는  $n$ -mark Golomb Ruler
  - $\{0, 1, 8, 12, 14, 17\}$ : 길이 17, 최적 6-mark Golomb Ruler
- 최적 Golomb Ruler를 이용한 exponent 행렬 생성[3]
  - 곱셈표의 상단행에  $n$ -mark Golomb Ruler의 원소들을 오름차순으로 배치하고 좌측열에  $(1, 2, 3)$ 을 배치하여 생성한  $3 \times n$  곱셈표를 exponent 행렬로 간주

×	0	1	8	12	14	17
1	0	1	8	12	14	17
2	0	2	16	24	28	34
3	0	3	24	36	42	51



# Girth Property

## Theorem [3]

The QC-LDPC codes from exponent matrix constructed from Golomb Ruler have girth 8 if

$$P > 2L,$$

where  $P$  is the modulus in the construction of the exponent matrix and  $L$  is the length of the Golomb ruler.

- Ex) 6-mark optimal Golomb ruler로 QC-LDPC 부호를 설계하는 경우
- CPM size  $P \geq 35$  로 설정 시 girth-8 QC-LDPC 부호를 설계 할 수 있음.

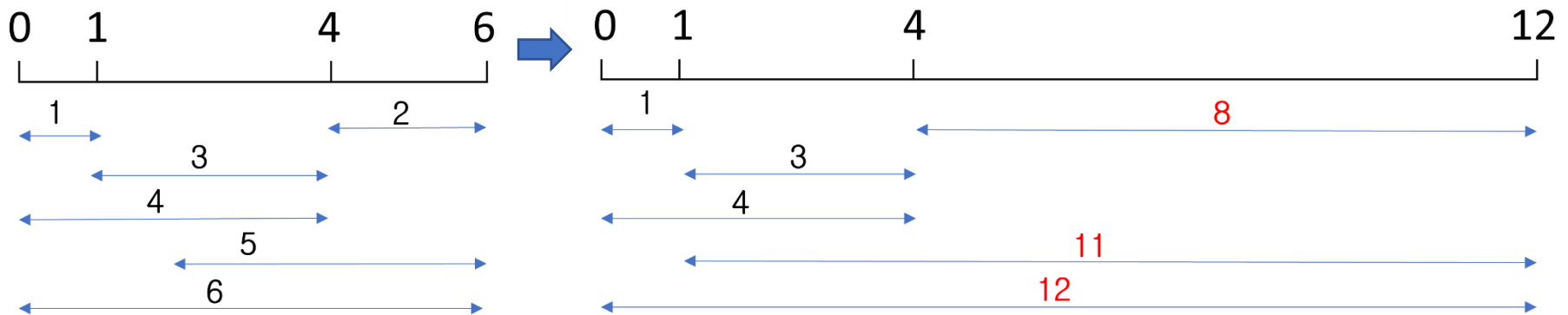
$\times$	0	1	8	12	14	17
1	0	1	8	12	14	17
2	0	2	16	24	28	34
3	0	3	24	36	42	51

[3] I. Kim and H.-Y. Song, "A construction for girth-8 QC-LDPC codes using Golomb rulers," accepted for publication, Electronic Letters, Mar. 2022.

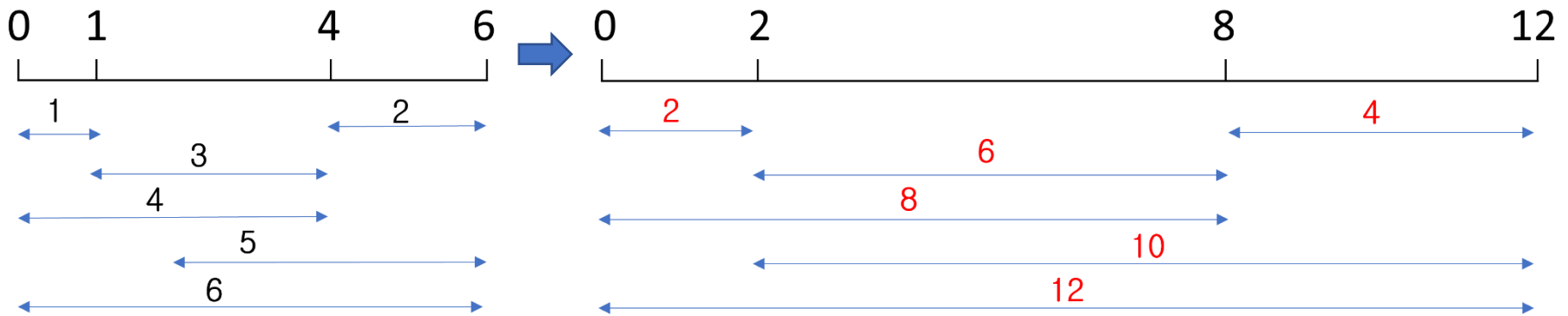


# 새로운 Golomb Ruler 생성법

## • Change-only-last 생성법



## • Change-all 생성법





# 새로운 Golomb Ruler 생성법

- Change-only-last 생성법

X 1:	0, 1, 8, 12, 14, 17
X 2:	0, 1, 8, 12, 14, 34
X 3:	0, 1, 8, 12, 14, 51
X 4:	0, 1, 8, 12, 14, 68
X 5:	0, 1, 8, 12, 14, 85

- Change-all 생성법

X 1:	0, 1, 8, 12, 14, 17
X 2:	0, 2, 16, 24, 28, 34
X 3:	0, 3, 24, 36, 42, 51
X 4:	0, 4, 32, 48, 56, 68
X 5:	0, 5, 40, 60, 70, 85



# 시뮬레이션 방식

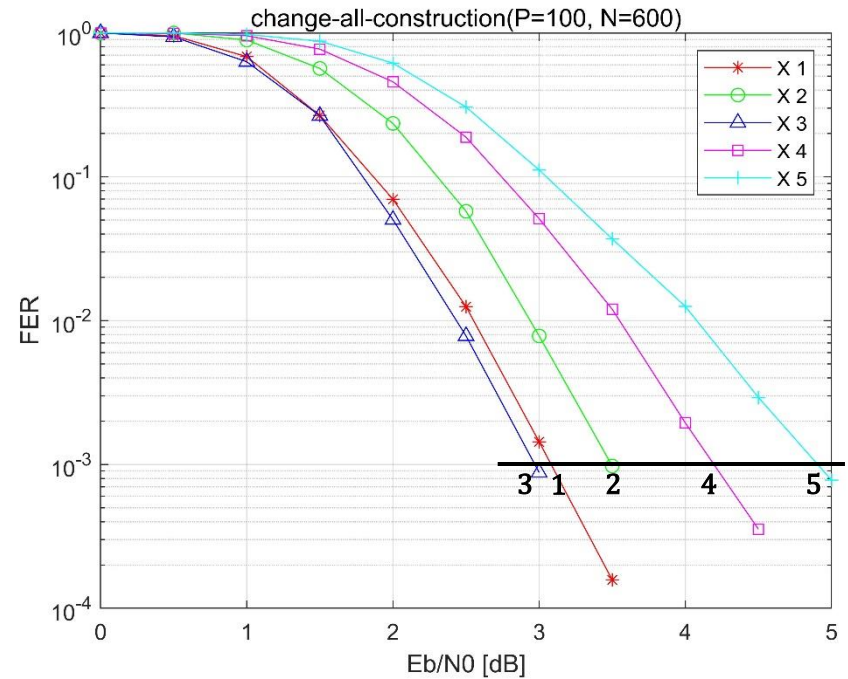
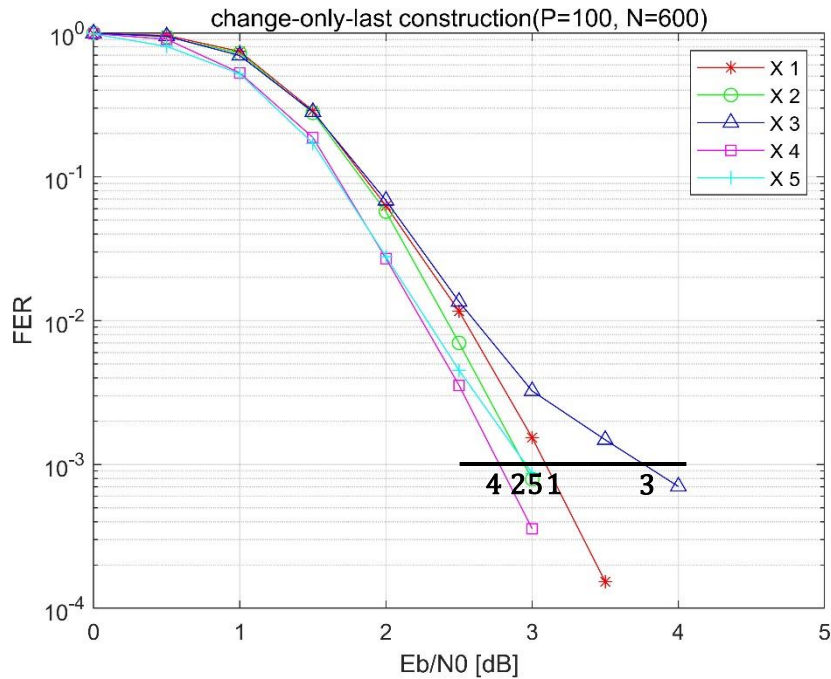
- LDPC 부호
  - 앞서 제시된 두 생성법으로 만든 10개의 Golomb Ruler들을 이용해 exponent 행렬들을 생성
  - CPM 사이즈  $P = 100, 200 \Rightarrow$  부호길이 600, 1200
- AWGN 채널 및 BPSK 변조 가정
- Sum-product 복호 방식
  - 최대 iteration : 50회
  - 최대 프레임 에러 : 200개



# P = 100(N=600)일 때 두 생성법의 성능 비교

## • FER $10^{-3}$ 성능비교

- change-only-last 생성법의 경우 × 4, × 2, × 5, × 1, × 3 순으로 성능 우수
- change-all 생성법의 경우 × 3, × 1, × 2, × 4, × 5 순으로 성능 우수
- 각 생성법의 best를 비교시 change-only-last 생성법이 change-all 생성법보다 약 0.1dB 우수

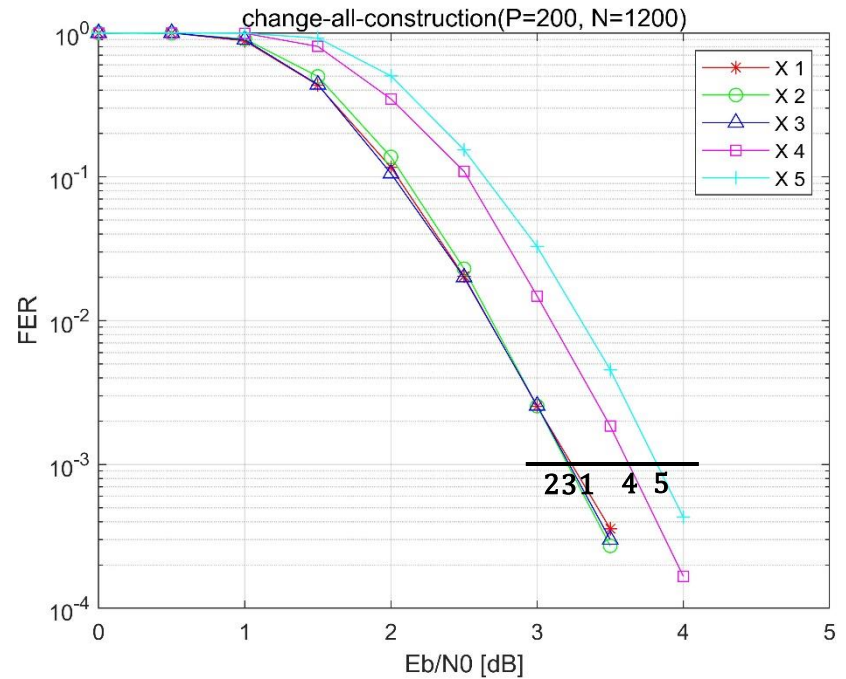
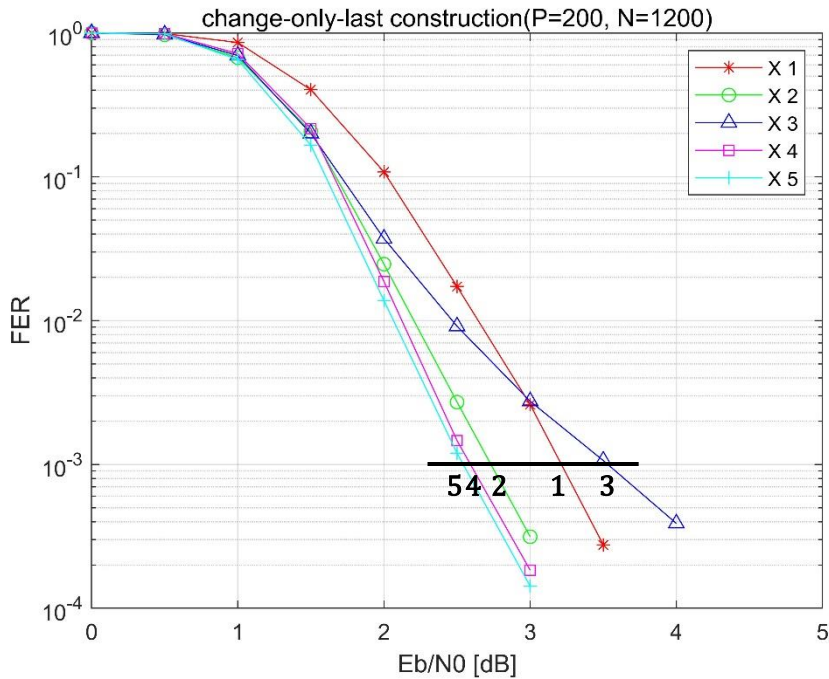




# P = 200(N=1200)일 때 두 생성법의 성능 비교

## • FER $10^{-3}$ 성능비교

- change-only-last 생성법의 경우  $\times 5, \times 4, \times 2, \times 1, \times 3$  순으로 성능 우수
- change-all 생성법의 경우  $\times 2, \times 3, \times 1, \times 4, \times 5$  순으로 성능 우수
- 각 생성법의 best를 비교시 change-only-last 생성법이 change-all 생성법보다 약 0.7dB 우수





## 결론

- change-only-last 및 change-all기법은 곱하는 수가 커짐에 따라 성능이 좋아지다 나빠지는 것을 확인
- 각 생성법의 best끼리 비교시 change-only-last 생성법이 change-all 생성법보다 우수한 성능을 보임
  - $P = 100$ 에서 change-only-last 생성법이 change-all보다 약 0.1dB 우수
  - $P = 200$ 에서 change-only-last 생성법이 change-all보다 약 0.7dB 우수
- 이러한 성능차이를 보이는 이유에 대해선 추가분석이 필요





## 참고자료

- 1) "Golomb ruler From Wikipedia, the free encyclopedia," Dec. 2021, Accessed on Jan. 11, 2022 [online] [https://en.wikipedia.org/wiki/Golomb\\_ruler](https://en.wikipedia.org/wiki/Golomb_ruler).
- 2) G. S. Bloom and S. W. Golomb, "Applications of numbered undirected graphs," *Proceedings of the IEEE*, vol. 65, no. 4, pp. 562-570, Apr. 1977.
- 3) I. Kim and H.-Y. Song, "A construction for girth-8 QC-LDPC codes using Golomb rulers," accepted for publication, *Electronic Letters*, Mar. 2022.
- 4) M. P. C. Fossorier, "Quasicyclic Low-density Parity-check Codes from Circulant Permutation Matrices," *IEEE Trans. Info. Theory*, vol. 50, pp. 1788-1793, July 2004.
- 5) R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja and D. J. Costello, "LDPC Block and Convolutional Codes Based on Circulant Matrices," *IEEE Trans. on Info. Theory*, vol. 50, no. 12, pp. 2966-2984, Dec. 2004.
- 6) I. Kim and H.-Y. Song, "Some New Constructions of Girth-8 QC-LDPC Codes for Future GNSS," *IEEE Communications Letters*, vol. 25, no. 12, pp. 3780-3784, Dec. 2021.