

# SC-LDPC 부호의 슬라이딩 윈도우 복호를 위한 곡선형 비균일 스케줄

채 상 원\*, 최 효 정\*, 김 강 산\*, 송 홍 엽<sup>o</sup>

## Curved Non-Uniform Schedules for Sliding Window Decoding of SC-LDPC Codes

Sangwon Chae\*, Hyojeong Choi\*, Gangsan Kim\*, Hong-Yeop Song<sup>o</sup>

요 약

본 논문은 공간 결합 저밀도 패리티 검사(Spatially-Coupled Low-Density Parity-Check, SC-LDPC) 부호의 슬라이딩 윈도우 복호 (Sliding Window Decoding, SWD)에서 성능-복잡도 간의 관계를 개선하기 위해 새로운 곡선형 비균일 스케줄을 제안한다. 기존의 선형(Straight) Pragmatic 스케줄은 Uniform 스케줄 대비 복잡도가 감소하지만, 반복 횟수가 적을 때 BER 성능 또한 저하된다. 이에 본 논문에서는 오류 정정 성능과 복호 복잡도 간의 균형을 유연하게 조절할 수 있는 볼록형 곡선과 오목형 곡선 기반의 두 가지 비선형 스케줄을 설계한다. 두 가지 SC-LDPC 부호 크기(10000×5100, 100000×51000)와 다양한 윈도우 크기(W=8, 10, 12)를 사용한 시뮬레이션을 통해, 볼록 곡선형 비균일 스케줄은 적은 반복 횟수에서도 Uniform 스케줄 대비 복잡도를 줄이면서 BER 성능을 거의 유지하는 반면, 오목 곡선형 비균일 스케줄은 성능 저하가 상당하지만, 복잡도가 가장 낮음을 확인한다.

**Key Words** : LDPC Codes, SC-LDPC Codes, Sliding Window Decoding.

### ABSTRACT

This paper proposes a new curved non-uniform schedule to improve the performance-complexity tradeoff of sliding window decoding (SWD) in spatially-coupled low-density parity-check (SC-LDPC) codes. The existing straight Pragmatic schedule reduces complexity compared to Uniform schedule, but also reduces BER performance when the number of iterations is low. This paper designs two curved schedules using convex and concave curves to flexibly adjust the balance between error correction performance and decoding complexity. Simulations were conducted using two SC-LDPC code sizes (10000×5100 and 100000×51000) and various window sizes (W=8, 10, 12). The results show that the convex curved non-uniform schedule maintains BER performance while reducing complexity compared to Uniform schedule even with fewer iterations, whereas the concave curved non-uniform schedule exhibits significant performance degradation but has the lowest complexity.

※ 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.RS-2023-00209000)

♦ First author : Yonsei University, Dep. Electrical and Electronic Engineering, sw.chae@yonsei.ac.kr, 학생회원

° Corresponding Author : Yonsei University, Dep. Electrical and Electronic Engineering, hysong@yonsei.ac.kr, 종신회원

\* Yonsei University, Dep. Electrical and Electronic Engineering, hjchoi3022@yonsei.ac.kr, 학생회원; gs.kim@yonsei.ac.kr, 정회원  
논문번호 : 202507-169-B-RN, Received July 23, 2025; Revised August 11, 2025; Accepted August 12, 2025

## I. 서 론

저밀도 패리티 검사(Low-Density Parity-Check, LDPC) 부호는 1962년 Gallager가 처음 제안한 선형 블록 오류 정정 부호로<sup>[1-3]</sup>, 1990년대 Mackay와 Neal에 의해 재발견된 이후 현대 통신시스템의 핵심 기술로 채택되어 사용되고 있다<sup>[4-7]</sup>.

공간 결합 LDPC(Spatially-Coupled LDPC, SC-LDPC) 부호는 여러 개의 작은 LDPC 부호를 체인(chain) 형태로 연결한 구조를 갖는 부호로, threshold saturation 현상을 통해 우수한 성능을 달성할 수 있다<sup>[8,9]</sup>. 하지만 전체 부호 복호 시 메모리 요구량과 계산 복잡도가 상당하여 실제 하드웨어 구현에 어려움이 있다. 이를 해결하기 위해 슬라이딩 윈도우 복호(Sliding Window Decoding, SWD) 기법이 제안되었다. SWD는 전체 부호를 동시에 복호하는 대신, 고정된 크기의 작은 윈도우 단위로, 차례대로 복호를 수행함으로써 하드웨어 복잡도와 지연시간을 크게 줄일 수 있다. 그러나 전체 부호 정보를 동시에 활용하지 못하므로 전체 블록 복호 대비 성능 손실이 발생한다<sup>[10]</sup>.

SWD의 성능을 유지하면서 복잡도를 더욱 줄이기 위해 다양한 개선 기법들이 제안되었다<sup>[11,12,13,14,15]</sup>. 그 중 Ul Hassan 등<sup>[14]</sup>은 2017년 비균일(Non-Uniform) 스케줄링 기법과 Pragmatic 스케줄을 제안하였다. Pragmatic 스케줄은 윈도우 내 노드 위치에 따라 선행적으로 업데이트 빈도를 결정하는 간단하면서도 효과적인 방법이다.

본 논문은 Ul Hassan 등이 제안한 기존의 Pragmatic 스케줄에 대비하여 SC-LDPC 부호의 복호 효율성을 높이기 위해 오목(Concave)과 볼록(Convex) 곡선 기반의 새로운 곡선형 비균일 스케줄을 제안한다. 이를 통해 SC-LDPC 부호의 SWD에서 복호 효율성과 성능 간의 새로운 균형점을 찾는 것을 목표로 한다.

본 논문의 II 장에서는 SC-LDPC 부호의 윈도우 복호 기법과 비균일 스케줄링 기법들을 소개한다. III 장에서는 본 논문에서 제안하는 곡선형 비균일 스케줄을 설명하고 설계 절차 및 복호 복잡도를 분석한다. IV 장에서는 실험을 통해 성능 평가 결과를 상세히 논의하고, V 장에서 주요 결과를 요약하며 결론을 맺는다.

## II. SC-LDPC 부호의 Pragmatic 스케줄

### 2.1 슬라이딩 윈도우 복호 알고리즘

LDPC 부호는 패리티 검사 행렬(parity-check matrix)  $H$ 에서 0이 아닌 원소의 개수가 전체 원소 대비

매우 적은 희소(sparse) 행렬 구조를 가진다. LDPC 부호의 복호는 주로 반복적 메시지 전달(iterative message passing) 알고리즘을 통해 이루어진다.

SC-LDPC 부호는 여러 개의 LDPC 부호를 결합(coupling)하여 구성된 구조적 부호이다. 기본 아이디어는 인접한 LDPC 부호들을 결합 길이(coupling width)  $w$ 만큼 겹치도록 연결하여 전파 효과(propagation effect)를 유도하고 수렴 성능을 높이는 것이다. 이러한 결합 구조는 threshold saturation 현상을 통해 Shannon 한계에 근접하는 성능을 달성할 수 있다<sup>[9]</sup>.

슬라이딩 윈도우 복호(Sliding-Window Decoding, SWD)는 SC-LDPC 부호의 구조적 특성을 활용하여 효율적으로 복호하는 기법이다. SWD는 하나의 고정된 크기  $W$ 의 윈도우(window) 내에서 국소적인 복호를 반복적으로 수행하고, 일정한 간격으로 윈도우를 이동시키며, 이를 반복하는 방식이다.

SWD의 장점으로는 전체 부호를 기다릴 필요 없이 일부 블록만으로 복호를 시작할 수 있고, 작은 크기의 윈도우에서 국소적인 연산만 수행하므로 하드웨어 자원 소모가 적고, 복호기의 구현이 단순해진다.

단점으로는 전체 부호를 동시에 고려하지 못하므로, 전체 블록 복호에 비해 약간의 성능 손실이 발생할 수 있다<sup>[20]</sup>. 이를 보완하기 위해 윈도우 크기를 적당히 크게 설정하거나, 다양한 스케줄링 기법을 활용할 수 있다.

### 2.2 비균일 스케줄 (Non-Uniform Schedule)

비균일 스케줄은 슬라이딩 윈도우 복호 기법에서 윈도우 내 각 검사 노드의 업데이트 횟수를 비균일하게 조정하는 방식으로, 중요도가 높은 검사 노드만 선택적으로 업데이트하여 복잡도를 절감한다<sup>[14]</sup>.

그중 [14]에서 소개한 BER 추정 기반 비균일 스케줄은 검사 노드의 중요도를 평가하기 위해, 각 검사 노드가 연결된 변수 노드의 상태를 분석하고, 이를 기반으로 가중치를 부여한다. 이 방식은 기존의 균일한 스케줄링 방식에 비해 복호 성능이 거의 유지되면서도 복호 복잡도를 크게 줄일 수 있으나 반복마다 노드의 중요도를 평가하는 추가적인 계산이 필요하다는 단점이 있다.

### 2.3 Pragmatic 스케줄

Pragmatic 스케줄은 비균일 스케줄의 한 가지 예로, 선형(Straight) 가중치 함수만을 사용하여 노드의 업데이트를 결정한다<sup>[14]</sup>.

기존의 Pragmatic 스케줄의 핵심 아이디어는 다음과 같은 간단한 규칙에 기반을 둔다. 윈도우 크기를  $W$ 로, 윈도우 내부 복호 횟수를  $T$ 로 나타낸다.

1. 반복  $i$ 에서 업데이트할 노드 위치(인덱스)의 최대값을  $w_{\max}(i)$ 로 정의한다( $i = 1, \dots, T$ ).
2. 첫 번째 반복에서는 윈도우 내 모든 노드를 업데이트한다( $w_{\max}(1) = W$ ).
3. 이후  $T-1$ 번의 반복에서는 반복마다 업데이트 영역이 하나씩 줄어든다( $w_{\max}(i) = W-i+1$ ,  $i = 2, 3, \dots, T$ ).
4. 총  $T$ 번의 반복 후,  $w_{\max}$ 는 다시  $W$ 로 초기화되고 최대 반복 횟수  $I_{\max}$ 에 도달할 때까지 이 과정을 반복한다.

여기서 2-3의 반복 과정을 하나의 주기(period)로 볼 수 있다.

이러한 방식은 반복 횟수에 따라 고정된 패턴을 사용하므로, 반복 과정 중에 추가적인 중요도 계산이 필요하지 않다는 장점이 있다. 이를 통해 복호 성능의 많은 감소 없이도 복잡도를 더욱 줄일 수 있다. 그림 1은 윈도우 크기  $W=8$ , 최대 반복 횟수  $I_{\max} = 16$ 의 예시이다.

기존 Pragmatic 스케줄의 선형적 업데이트 패턴 외

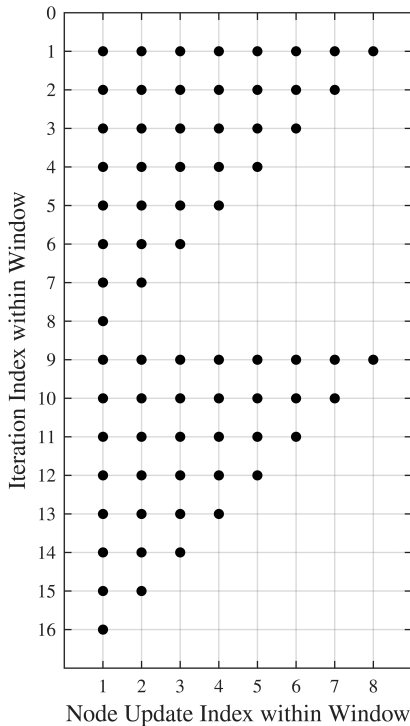


그림 1. 기존 Pragmatic 스케줄의 윈도우 크기 8, 최대 반복 횟수 16의 예시  
Fig. 1. Example of an existing Pragmatic schedule with a window size of 8 and a maximum iteration number of 16.

에 다른 형태의 스케줄링 방식을 통해 성능-복잡도 간의 관계를 더욱 세밀하게 조절할 가능성을 탐구하기 위해, 3장에서 곡선형 비균일 스케줄을 제안한다.

### III. 새로운 비균일 스케줄 제안 및 실험

#### 3.1 곡선형 비균일 스케줄 설계

본 연구에서는 기존의 Pragmatic 스케줄을 확장하여 곡선형 비균일 스케줄을 제안한다. 곡선형 스케줄은 복호 복잡도와 성능 간의 균형을 조절할 수 있는 스케줄링 기법이다.

곡선형 스케줄의 설계 방식은 다음과 같다.

- 전체 스케줄: 최대 반복 횟수에 도달할 때까지 여러 개의 주기(period)로 구성된다. 본 실험에서는 최대 반복 횟수로 2주기( $I_{\max} = 2W$ )를 사용한다.
- 단일 주기: 각 주기(period)는 윈도우 크기  $W$ 만큼의 반복 횟수를 가진다. 즉, 각 주기는  $W$ 번의 반복(iteration)으로 구성된다.
- 곡선 설계: 단일 주기에서 반복 횟수에 따른 노드의 업데이트 영역을 곡선 함수로 정의하여 결정한다.

업데이트 영역은 노드 위치(index)와 반복 횟수를 모두 0부터 1 사이의 정규화된 값으로 변환하여 다룬다.

- $x$  좌표: 윈도우 내의 노드 위치  $j$ 는 1부터  $W$ 까지의 값을 가지며, 이를  $[0,1)$  구간으로 정규화하면 수식 1과 같다.

$$x = \frac{j-1}{W}, \quad x \in [0, 1). \quad (1)$$

- $y$  좌표: 반복 횟수  $i$  역시 1부터  $W$ 까지의 값을 가지며, 이를  $[0,1)$  구간으로 정규화하면 수식 2와 같다.

$$y = \frac{i-1}{W}, \quad y \in [0, 1) \quad (2)$$

- 정규화된 노드 위치  $x$ 에서 곡선의 값이 현재 반복의 정규화된 값  $y$ 보다 크거나 같다면 해당 노드 위치를 업데이트 영역에 포함한다.

곡선형 스케줄 설계의 핵심 원리는 다음과 같다. 첫째, SWD에서 복호 대상은 항상 가장 왼쪽 변수 노드 블록이므로, 가장 왼쪽 노드 위치를 항상 업데이트한다. 둘째, 기존 선형 스케줄과의 명확한 비교를 위해 모든 노드 위치에서 선형 스케줄 대비 일관적으로 많거나 적

은 업데이트 횟수를 갖도록 설계하였다. 이러한 설계 원리에 따라 본 연구에서는 1/4 원호를 기반으로 한 블록 및 오목 곡선을 대표 곡선으로 선정했다.

### 3.1.1 블록(Convex) 곡선형 비균일 스케줄

블록(Convex) 곡선형 비균일 스케줄은 선형 스케줄보다 상대적으로 많은 수의 노드를 업데이트한다. 본 연구에서 선택한 블록 곡선형 비균일 스케줄(Convex)의 업데이트 영역은 수식 3과 같다.

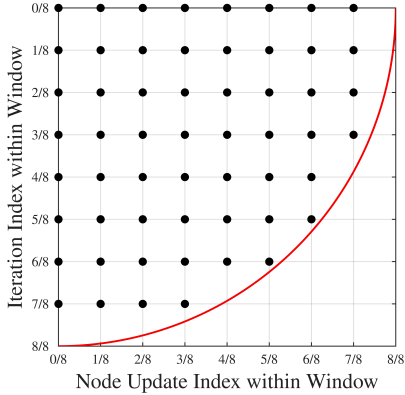


그림 2. W=8, 블록 곡선형 비균일 스케줄 설계 예시  
Fig. 2. W=8, convex curved non-uniform schedule

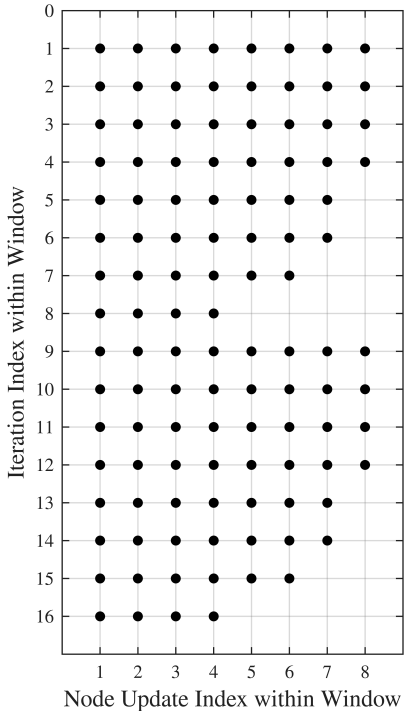


그림 3. W=8, 블록 곡선형 비균일 스케줄 설계 예시  
Fig. 3. W=8, convex curved non-uniform schedule

$$y \leq \sqrt{1-x^2} \quad \text{for } 0 \leq x < 1 \quad (3)$$

그림 2는  $W=8$ , 단일 주기의 설계 예시이다. 본 연구에서는 그림 3과 같이 두 주기로 실험을 진행했다.

### 3.1.2 오목(Concave) 곡선형 비균일 스케줄

오목(Concave) 곡선형 비균일 스케줄은 선형 스케줄보다 상대적으로 적은 수의 노드를 업데이트한다. 본 연구에서 선택한 오목 곡선형 비균일 스케줄의 업데이트

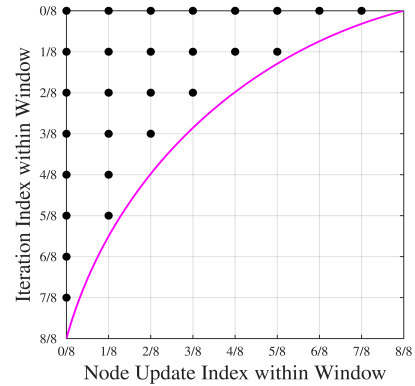


그림 4. W=8, 오목 곡선형 비균일 스케줄 설계 예시  
Fig. 4. W=8, concave curved non-uniform schedule

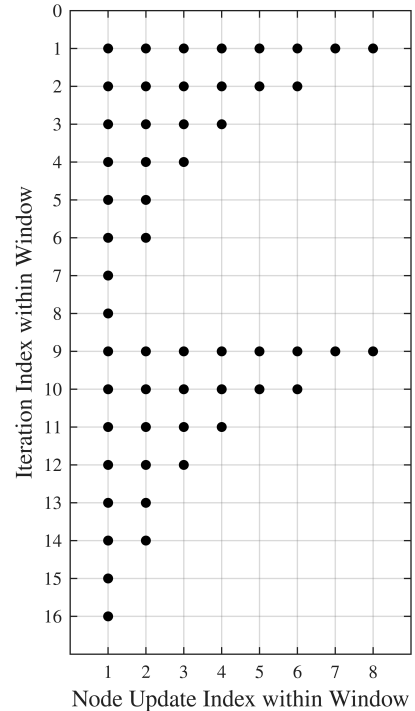


그림 5. W=8, 오목 곡선형 비균일 스케줄 설계 예시  
Fig. 5. W=8, concave curved non-uniform schedule

트 영역은 수식 4과 같다.

$$y \leq b - \sqrt{2 - (x - b)^2}, b = \frac{1 + \sqrt{3}}{2} \quad \text{for } 0 \leq x < 1 \quad (4)$$

그림 4는  $W=8$ , 단일 주기의 설계 예시이다. 본 논문에서는 그림 5와 같이 두 주기로 실험을 진행했다.

#### IV. 실험 결과

본 장에서는 두 종류의 SC-LDPC 부호에 대해 제안된 곡선형 비균일 스케줄의 성능 평가 결과를 기존의 SWD(Uniform) 및 선형 Pragmatic 스케줄과 비교하여 분석한다. 비트 오류율(BER), 복잡도, 그리고 다양한 윈도우 크기에 따른 성능-복잡도 관계를 중심으로 분석 결과를 제시한다.

본 실험에서는 Additive White Gaussian Noise (AWGN) 채널을 가정하고, Binary Phase Shift Keying (BPSK) 변조 방식을 사용하였다.

실험에 사용된 SC-LDPC 부호는 프로토 그래프 기반 설계 방식을 사용하였으며, 사용된 파라미터들은 표 1과 같다.

윈도우 크기(window size)  $W$ 는 실험적으로 다양한 값(8, 10, 12)을 설정하여 성능 및 복잡도에 미치는 영향을 분석하였다. SWD의 윈도우 내부 복호 알고리즘으로 Sum-Product를 사용했으며, 반복 횟수는 윈도우 크기의 2배인  $2W$  (2주기)로 고정했다.

표 1. SC-LDPC 부호 파라미터  
Table 1. SC-LDPC code parameters

파라미터 종류	부호 1	부호 2
에지-스프레딩 이후 기반 행렬 (base matrix)	$\mathbf{B} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$	
결합 길이 (coupling width)	$w = 2$	
프로토그래프 개수	$L = 100$	$L = 100$
리프팅 계수 (lifting factor $z$ )	$z = 50$	$z = 500$
패리티 검사 행렬 크기	10000 × 5100	100000 × 51000
부호율(code rate)	0.49	0.49

##### 4.1 BER 성능 비교

본 절에서는 두 종류의 SC-LDPC 부호에 대해 다양

한 윈도우 크기에 따라 제안된 곡선형 비균일 스케줄의 비트 오류율(BER)을 평가한다.

그림 6-8은 10000×5100 크기의 SC-LDPC 부호에서 스케줄에 따른 성능을 윈도우의 크기를  $W=8, 10, 12$ 로 변화를 주며 비교한 결과다.

10000×5100 크기의 SC-LDPC 부호에서  $\text{BER } 10^{-5}$  기준을 달성할 수 있는  $E_b/N_0$  값을 표 2에서 비교하였다.

그림 9-11은 100000×51000 크기의 SC-LDPC 부호에서 스케줄링에 따른 성능을 윈도우의 크기를  $W=8, 10, 12$ 로 바꿔가면서 비교한 결과다.

100000×51000 크기의 SC-LDPC 부호에서  $\text{BER } 10^{-5}$  기준을 달성할 수 있는  $E_b/N_0$  값을 표 3에서 비교하였다.

전반적으로 Convex 곡선 스케줄은 Concave 곡선 및 선형 스케줄 대비 우수한 BER 성능을 보인다. 특히 주목할 점은 Convex 곡선 스케줄이 Uniform 스케줄과 거의 동일한 BER 성능을 달성한다는 것이다. 한편, 이러한 성능 차이는 윈도우 크기  $W$ 가 커질수록 감소하는 경향을 보인다.

표 2. (10000, 5100) SC-LDPC 부호에서  $\text{BER } 10^{-5}$ 를 위한  $E_b/N_0$   
Table 2.  $E_b/N_0$  for  $\text{BER } 10^{-5}$  in (10000, 5100) SC-LDPC code

$E_b/N_0$ [dB]	$W=8$	$W=10$	$W=12$
Non-Uniform, Concave	3.89	3.57	3.40
Non-Uniform, Straight	3.48	3.12	2.70
Non-Uniform, Convex	3.21	2.70	2.48
Uniform	3.19	2.70	2.45

표 3. (100000, 51000) SC-LDPC 부호에서  $\text{BER } 10^{-5}$ 를 위한  $E_b/N_0$   
Table 3.  $E_b/N_0$  for  $\text{BER } 10^{-5}$  in (100000, 51000) SC-LDPC code

$E_b/N_0$ [dB]	$W=8$	$W=10$	$W=12$
Non-Uniform, Concave	2.20	1.76	1.66
Non-Uniform, Straight	1.72	1.55	1.46
Non-Uniform, Convex	1.61	1.48	1.40
Uniform	1.61	1.47	1.38

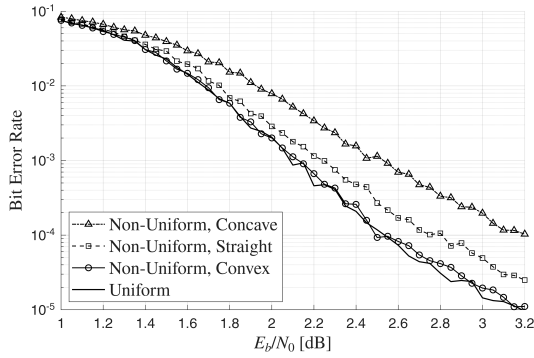


그림 6. W=8 (10000, 5100) SC-LDPC 부호의 서로 다른 스케줄에 따른 BER 성능  
Fig. 6. BER performance of different schedules for W=8 (10000, 5100) SC-LDPC codes

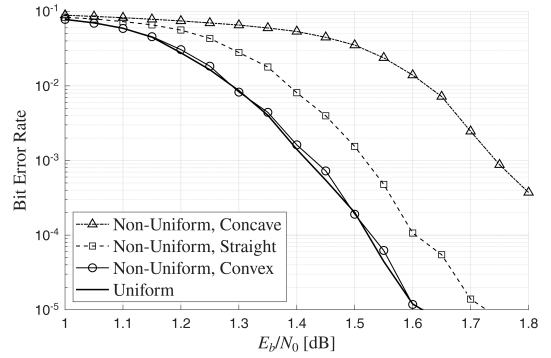


그림 9. W=8 (100000, 51000) SC-LDPC 부호의 서로 다른 스케줄에 따른 BER 성능  
Fig. 9. BER performance of different schedules for W=8 (100000, 51000) SC-LDPC codes

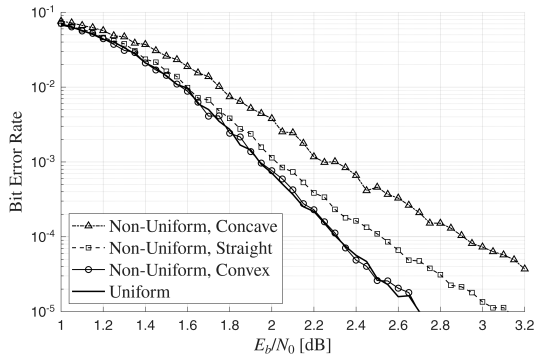


그림 7. W=10 (10000, 5100) SC-LDPC 부호의 서로 다른 스케줄에 따른 BER 성능  
Fig. 7. BER performance of different schedules for W=10 (10000, 5100) SC-LDPC codes

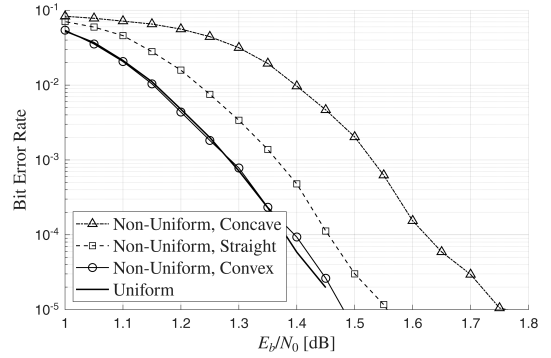


그림 10. W=10 (100000, 51000) SC-LDPC 부호의 서로 다른 스케줄에 따른 BER 성능  
Fig. 10. BER performance of different schedules for W=10 (100000, 51000) SC-LDPC codes

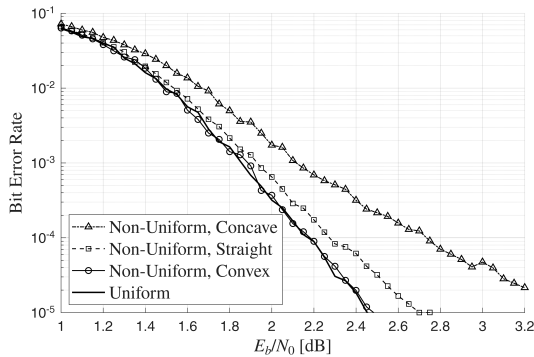


그림 8. W=12 (10000, 5100) SC-LDPC 부호의 서로 다른 스케줄에 따른 BER 성능  
Fig. 8. BER performance of different schedules for W=12 (10000, 5100) SC-LDPC codes

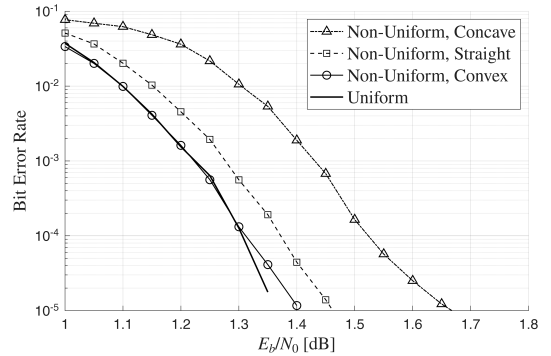


그림 11. W=12 (100000, 51000) SC-LDPC 부호의 서로 다른 스케줄에 따른 BER 성능  
Fig. 11. BER performance of different schedules for W=12 (100000, 51000) SC-LDPC codes

#### 4.2 복호 복잡도 비교

복호 복잡도는 평균 노드 업데이트 횟수  $U_{avg}$ 로 비교하였다<sup>[4]</sup>. 본 논문에서 제안하는 곡선형 비균일 스케

줄은 Ul Hassan 등<sup>[4]</sup>의 비균일 스케줄과 같이 특정 인덱스에서 변수 노드와 검사 노드 업데이트를 함께 수행하거나 함께 생략되기에 상대적 복잡도 지표로 볼 수 있다.

표 4. SC-LDPC 부호에서 서로 다른 스케줄에 따른 평균 노드 업데이트 횟수

Table 4. Average number of node updates under different schedules in SC-LDPC codes.

$U_{avg}$	$W=8$	$W=10$	$W=12$
Non-Uniform, Concave	52.92	81.82	110.44
Non-Uniform, Straight	70.32	106.70	150.28
Non-Uniform, Convex	108.50	168.88	230.40
Uniform	123.52	191.00	272.16

표 4는 두 SC-LDPC 부호에서 서로 다른 스케줄에 따른 평균 노드 업데이트 횟수를 비교한 결과이다. 두 SC-LDPC 부호에서 LDPC 프로그래밍 개수 혹은 윈도우 위치의 개수가 같으므로 평균 노드 업데이트 횟수  $U_{avg}$ 는 동일하다.

스케줄 별로 복잡도를 비교하면 Concave 곡선 스케줄은 가장 낮은 복잡도를 보이며 Straight 스케줄 대비 약 23-27%, Uniform 스케줄 대비 약 57-59% 낮다. 주목할 점은 Convex 곡선 스케줄이 Uniform 스케줄 대비 약 12-15%의 복잡도 감소를 달성하면서도 BER 성능을 거의 유지한다는 것이다. 반면 Straight 스케줄은 Uniform 대비 약 43-45%의 복잡도 감소를 보이지만 BER 성능 저하를 수반한다.

윈도우 크기에 따라서 복잡도 역시 증가하는데,  $W=8$ 에서  $W=10$ 으로 증가 시 복잡도가 약 51-55% 증가하며,  $W=10$ 에서  $W=12$ 로 증가 시 복잡도가 약 34-42% 증가한다.

#### 4.3 성능-복잡도 비교

본 절에서는 다양한 윈도우 크기와 스케줄링 방식의 조합이 BER 성능과 복잡도에 미치는 영향을 분석한다.

그림 12는 (100000, 51000) SC-LDPC 부호에 대한 성능-복잡도 그래프를 보여준다. 가로축은 평균 노드 업데이트 횟수  $U_{avg}$ , 세로축은 BER  $10^{-5}$  기준 요구  $E_b/N_0$ 이다. 윈도우 크기  $W$ 를 마커 옆 숫자로 표기하였다. 스케줄 별로 선을 그려 추세를 쉽게 볼 수 있게 했다. Concave 스케줄의 경우 BER 성능이 다른 스케줄에 비해 매우 낮아 비교 대상에서 제외하였다.

전체적으로, Straight가 복잡도가 가장 낮고 Convex, Uniform 순서로 복잡도가 높아지고, 성능도 개선되는 모습을 보인다. 모든 스케줄에서  $W$ 가 8에서 10으로, 10에서 12로 증가함에 따라  $U_{avg}$ 가 약 1.5배,

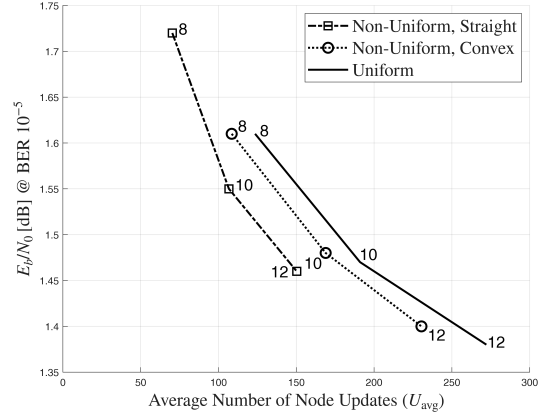


그림 12. (100000, 51000) SC-LDPC 부호의 성능-복잡도  
Fig. 12. Performance-Complexity of (100000, 51000) SC-LDPC code.

1.40배 증가하는 반면 필요  $E_b/N_0$ 는 약 0.15 dB, 0.09 dB 개선되었다.

또한 같은  $W$ 에서 Uniform과 Convex 스케줄의 필요  $E_b/N_0$  값은 큰 차이를 보이지 않지만, 더 적은 업데이트 횟수를 가진 Straight 스케줄은 같은 BER 성능을 위해 더 높은  $E_b/N_0$  값이 필요하다.

## V. 결 론

본 논문에서는 SC-LDPC 부호의 슬라이딩 윈도우 복호에서 기존 Uniform 스케줄과 선형(Straight) Pragmatic 스케줄 간의 성능-복잡도 관계를 개선하기 위해 오목(Concave)과 볼록(Convex) 곡선 기반의 새로운 곡선형 비균일 스케줄을 제안하였다.

두 가지 서로 다른 크기의 SC-LDPC 부호를 시뮬레이션한 결과, 가장 주목할 만한 성과는 제안한 Convex 곡선 스케줄이 Uniform 스케줄 대비 약 12-15%의 복잡도 감소를 달성하면서도 거의 동일한 BER 성능을 유지한다는 것이다. 이는 성능 저하 없이 복잡도를 줄일 수 있는 효과적인 방법을 제시한다. 또한 제안한 Concave 곡선 스케줄은 Straight 스케줄 대비 약 23-27%의 복잡도 감소를 달성하여 큰 복잡도 절감 효과를 보였다. 그리고 성능-복잡도 그래프 분석을 통해 각 스케줄의 효율성을 확인하였다.

본 연구는 SC-LDPC 부호의 복호 성능과 복잡도 간 균형에 대한 분석을 제공함으로써, 다양한 응용 환경에 맞는 최적의 복호 파라미터 선택에 이바지할 수 있다. 향후 연구에서는 제안된 비균일 스케줄링 방식의 다양한 채널 환경(페이딩, 심볼 간 간섭 등)에서의 성능 검

중, BPSK 외 다른 변조 기법에서의 성능 검증, 그리고 하드웨어 구현 및 성능 검증을 통해 제안된 기법의 실용성을 더욱 높일 수 있을 것으로 기대된다.

## References

- [1] R. Gallager, "Low-density parity-check codes," *IRE Trans. Inf. Theory*, vol. 8, no. 1, pp. 21-28, Jan. 1962.  
(<https://doi.org/10.1109/TIT.1962.1057683>)
- [2] S. Lin and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, 2nd ed., Pearson, 2004.
- [3] S. Lin and J. Li, *Fundamentals of Classical and Modern Error-Correcting Codes*, Cambridge University Press, 2021.
- [4] J. G. Proakis and M. Salehi, *Digital Communications*, McGraw-Hill Education, 2008.
- [5] *5G; NR; Multiplexing and channel coding (3GPP TS 38.212 version 18.5.0 Release 18)*, Jan. 2025.
- [6] *EN 302 307-2 - V1.4.1 - Digital Video Broadcasting (DVB); Second generation framing structure, channel coding and modulation systems for Broadcasting, Interactive Services, News Gathering and other broadband satellite applications; Part 2: DVB-S2 Extensions (DVB-S2X)*, Aug. 21, 2024.
- [7] *TM Synchronization and Channel Coding Recommended Standard Ccsds 131.0-B-5*, Sep. 2023.
- [8] A. J. Felstrom and K. S. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *IEEE Trans. Inf. Theory*, vol. 45, no. 6, pp. 2181-2191, Jan. 1999.  
(<https://doi.org/10.1109/18.782171>)
- [9] S. Kudekar, T. J. Richardson, and R. L. Urbanke, "Threshold saturation via spatial coupling: Why convolutional LDPC ensembles perform so well over the BEC," *IEEE Trans. Inf. Theory*, vol. 57, no. 2, pp. 803-834, Jan. 2011.  
(<https://doi.org/10.1109/TIT.2010.2095072>)
- [10] N. Ul Hassan, M. Lentmaier, and G. P. Fettweis, "Comparison of LDPC block and LDPC convolutional codes based on their decoding latency," in *2012 7th Int. Symp. Turbo Codes and Iterative Inf. Process. (ISTC)*, pp. 225-229, Gothenburg, Sweden, Aug. 2012.  
(<https://doi.org/10.1109/ISTC.2012.6325232>)
- [11] X. Jia, J. Li, Y. Yao, Y. Wang, B. Liu, and D. Xu, "Design of improved error-rate sliding window decoder for SC-LDPC codes: Reliable termination and channel value reuse," *Optoelectronics Lett.*, vol. 21, no. 4, pp. 212-217, Feb. 2025.  
(<https://doi.org/10.1007/s11801-025-4054-4>)
- [12] I. Ali, J.-H. Kim, S.-H. Kim, H. Kwak, and J.-S. No, "Improving windowed decoding of SC LDPC codes by effective decoding termination, message reuse, and amplification," *IEEE Access*, vol. 6, pp. 9336-9346, Nov. 2017.  
(<https://doi.org/10.1109/ACCESS.2017.2771375>)
- [13] L. Wei, T. Koike Akino, D. G. M. Mitchell, T. E. Fuja, and D. J. Costello Jr., "Threshold analysis of non binary spatially coupled LDPC codes with windowed decoding," in *2014 IEEE ISIT*, pp. 881-885, Honolulu, HI, USA, Jun.-Jul. 2014.  
(<https://doi.org/10.1109/ISIT.2014.6874959>)
- [14] N. U. Hassan, A. E. Pusane, M. Lentmaier, G. P. Fettweis, and D. J. Costello, "Non-uniform window decoding schedules for spatially coupled LDPC codes," *IEEE Trans. Commun.*, vol. 65, no. 2, pp. 501-510, Nov. 2016.  
(<https://doi.org/10.1109/TCOMM.2016.2633466>)
- [15] E. Ha, G. Kim, S. Chae, and H.-Y. Song, "Performance comparison of SC-LDPC codes and QC-LDPC codes with modified window decoding scheduling," in *Proc. Symp. KICS*, pp. 1246-1247, Jeju Island, Korea, Jun. 2024.
- [16] S. Khittiwitachayakul, W. Phakphisut, and P. Supnithi, "Integer programming based non-uniform window decoding schedules for spatially coupled low density parity check



codes,” *IET Commun.*, vol. 16, no. 17, pp. 2019-2035, Jul. 2022.

(<https://doi.org/10.1049/cmu2.12456>)

- [17] J. Kim, H.-G. Joo, and D.-J. Shin, “Effective bi-directional overlapped sliding window decoding of SC-LDPC codes,” *ICT Express*, vol. 10, no. 3, pp. 513-518, Nov. 2023. (<https://doi.org/10.1016/j.ict.2023.11.006>)

- [18] Q.-F. Lian, Q. Chen, L. Zhou, Y.-C. He, and X. Xie, “Adaptive decoding algorithm with variable sliding window for double SC-LDPC coding system,” *IEEE Commun. Lett.*, vol. 27, no. 2, pp. 404-408, Nov. 2022. (<https://doi.org/10.1109/LCOMM.2022.3222560>)

- [19] R. Tanner, “A recursive approach to low complexity codes,” *IEEE Trans. Inf. Theory*, vol. 27, no. 5, pp. 533-547, Sep. 1981. (<https://doi.org/10.1109/TIT.1981.1056404>)

- [20] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498-519, Jan. 2001. (<https://doi.org/10.1109/18.910572>)

- [21] J. Thorpe, “Low-density parity-check (LDPC) codes constructed from protographs,” *Interplanetary Network Progress Report*, pp. 1-7, Aug. 2003.

- [22] P. Hailes, L. Xu, R. G. Maunder, B. M. Al-Hashimi, and L. Hanzo, “A survey of FPGA-based LDPC decoders,” *IEEE Commun. Surv. & Tuts.*, vol. 18, no. 2, pp. 1098-1122, Dec. 2015. (<https://doi.org/10.1109/COMST.2015.2510381>)

#### 채 상 원 (Sangwon Chae)



2023년 2월 : 연세대학교 전기  
전자공학부 졸업

2023년 3월~현재 : 연세대학교  
전기전자공학과 석사과정

<관심분야> 통신공학, 정보이  
론, 부호이론

#### 최 효 정 (Hyojeong Choi)



2018년 2월 : 연세대학교 정보  
통신공학부 졸업

2021년 2월 : 연세대학교 전기  
전자공학과 석사

2021년 3월~현재 : 연세대학교  
전기전자공학과 박사과정

<관심분야> 통신공학, 부호이론,  
분산저장시스템, 카오스이론

[ORCID:0000-0003-2305-5111]

#### 김 강 산 (Gangsan Kim)



2016년 2월 : 연세대학교 전기  
전자공학부 졸업

2016년 3월~2025년 2월 : 연세  
대학교 전기전자공학과 석박  
통합과정

2022년 5월~2023년 10월 : 육  
군교육사령부 인공지능과학  
기술연구병

2025년 3월~현재 : 연세대학교 전기전자공학과 박사  
후연구원

<관심분야> 통신공학, 정보이론, 부호이론

[ORCID:0000-0002-3864-5379]

#### 송 홍 엽 (Hong-Yeop Song)



1984년 2월 : 연세대학교 전자  
공학과 졸업

1986년 5월 : University of  
Southern California Dept.  
of EE. System 석사

1991년 12월 : University of  
Southern California Dept.

of EE. System 박사

1992년 1월~1993년 12월 : University of Southern  
California 박사 후 연구원

1994년 1월~1995년 8월 : Qualcomm, San Diego,  
Senior Engineer

1995년 9월~현재 : 연세대학교 전기전자공학과 전임  
교수

<관심분야> 통신공학, 정보이론, 부호이론

[ORCID:0000-0001-8764-9424]