

LDPC Codes에서 PEG 알고리즘과 IPEG 알고리즘의 성능 비교 평가 및 분석

*김현승, *고재현, *장민[†], *김상호

*성균관대학교

minycloud@gmail.com[†]

Systematic Performance Analysis of the PEG and IPEG in the LDPC Codes

*Hyun-Seung Kim, *Jae-Hyun Ko, *Min Jang, *Sang-Hyo Kim

*Sungkyunkwan University

요약

1962년 R.G Gallager가 제안한 LDPC(Low Density Parity Check)부호는 Shannon의 채널 용량의 한계에 근접한 우수한 오류정정 부호이다. 우수한 LDPC부호 생성 조건 중 가장 중요한 부분은 바로 최소 사이클 길이(girth)를 최대화 하는 과정인데, PEG(Progressive Edge Growth)알고리즘은 이 조건을 만족시키는 우수한 알고리즘으로 인정받고 있다. 이후 높은 SNR범위에서 PEG알고리즘의 성능을 개선한 IPEG (Improved PEG) 알고리즘을 포함한 다양한 알고리즘이 제안 되었다.본 논문은 PEG와 IPEG 알고리즘을 이용해 생성한 LDPC 부호를 이용하여 부호길이, 부호율을 변화시키면서 실험하여 그 결과를 비교 분석하였고, 더 좋은 성능을 가질 수 있는 알고리즘에 대해 논의한다.

1. 서론

1962년 R.G Gallager가 제안한 LDPC(Low Density Parity- Check)부호는 Shannon의 채널 용량의 한계에 근접한 우수한 오류정정(error correction) 부호이다. 제안당시에는 LDPC 부호 생성의 복잡성 때문에 많은 주목을 받지 못했지만 IC(Integrated Circuit) 기술의 발전에 따른 고성능 컴퓨터의 개발과 더불어 효율적인 LDPC 부호 생성 알고리즘이 많이 개발되면서 활발한 연구가 진행되어 왔다.

우수한 LDPC부호 생성 조건 중 가장 중요한 것은 바로 girth(최소 사이클 길이)를 최대화 하는 과정이다. Xiao-Yu Hu와 Dieter M. Arnold 가 제안한 PEG(Progressive Edge-Growth)알고리즘은 이 조건을 만족시키는 우수한 알고리즘으로 인정받고 있다. 이후 Hua Xiao와 Amir H. Banihashemi는 높은 SNR 범위 에서 PEG알고리즘의 성능을 개선한 IPEG (Improved PEG) 알고리즘이 제안하였다. 그 외에 PEG 알고리즘을 기반으로 한 여러 다른 형태의 성능을 향상시킨 알고리즘이 제안되었다. 본 논문은 PEG와 IPEG 알고리즘으로 생성한 패리티검사 행렬을 이용하여 BPSK(Binary Phase Shift Keying 변조된 신호를 AWGN(Addictive White Gaussian Noise) 채널 환경에서 복호화하는 과정을 실험해서 BER(Bit Error Rate)관점에서 결과를 출력하고, 두 알고리즘의 결과를 비교 분석하였다.

2. PEG 알고리즘

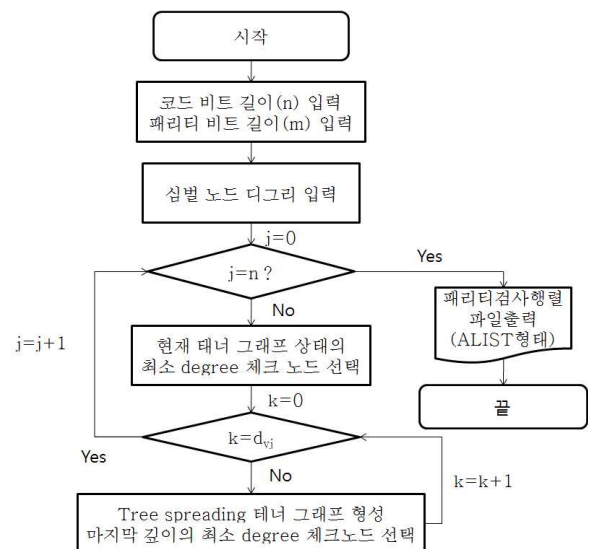
가. 개요

성능이 뛰어난 LDPC부호를 생성하기 위해서는 큰 girth를 갖는 tanner 그래프(Tanner Graph)를 구성해야 하는데 PEG 알고리즘을 통해 이 조건을 충분히 만족시킬 수 있다.

PEG알고리즘의 핵심은 현재 심벌노드 s_j 에서 가장 먼 거리에 있는 체크노드를 찾은 후에 해당 체크노드를 연결하여 새로운 엣지를 배치하는 것이다. 이 방법은 트리확장을 통해 구현할 수 있다. 가장 먼 거리에 있는 체크노드 집합 중에서 차수가 가장 작은 체크노드를 선택하고, 만약 최소 차수를 가진 체크노드가 2개 이상 존재한다면 임의로 한 체크노드를 선택한다. 이와 같은 방법을 통해 패리티검사행렬(Parity Check Matrix)의 small cycle 생성을 억제하고 국부 girth를 최대화할 수 있다. PEG알고리즘의 자세한 설명은 논문[1]에 기재되어 있다.

나. 순서도

앞서 설명한 PEG 알고리즘의 구현 방법을 순서도로 표현하였다.



< 그림1. PEG알고리즘 순서도 >

3. IPEG 알고리즘

가. 개요

PEG알고리즘의 오류마루현상을 개선한 IPEG 알고리즘이다. IPEG는 ACE(Approximate Cycle Extrinsic message 차수) 개념을 도입하여 위에서 설명한 PEG 알고리즘에서 발생하는 임의의 선택 조건을 개선시킨 알고리즘이다. ACE는 특정 노드를 포함하고 있는 순환노드에서 연결된 외부 순환노드와의 연관성을 의미한다. ACE가 클수록 외부 정보를 받고 내보내기 쉽기 때문에 정보가 순환노드안에 고립되지 않는다. 이로써 SPA(Sum Product Algorithm) 복호과정에서 오류율을 낮추는 효과를 준다. ACE는 다음과 같이 정의된다.

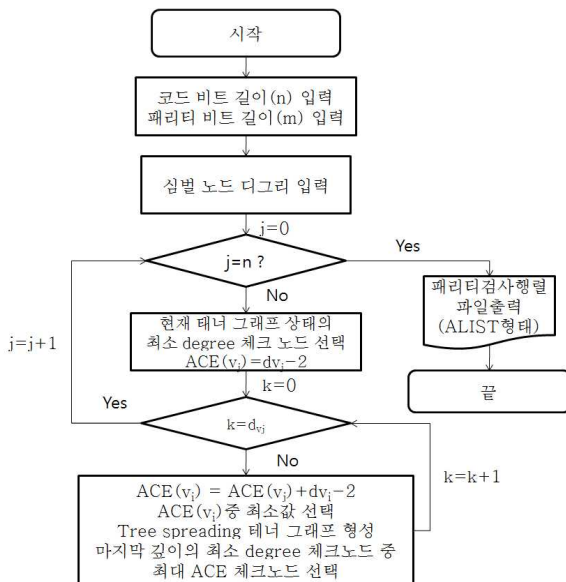
$$ACE(v_j) = d_{v_j} - 2 \quad v_j: j \text{ 번째 심벌노드}$$

$$ACE(c_i) = 0 \quad c_i: i \text{ 번째 체크노드}$$

ACE 검출 방법은 다음과 같다. 각각의 심벌노드와 체크노드는 이전 단계의 연결된 노드들로부터 전달 받은 ACE값들 중 가장 작은 것과 자기 자신의 ACE를 합한 값을 아랫단계의 연결된 노드들로 전달한다. 결국 마지막 단계의 체크노드에는 최소화된 ACE 값들을 가지고 있게 된다. ACE Detection 에 관한 상세한 설명은 [3]에 제시되어 있다. PEG 알고리즘에서는 멀리 있는 체크노드의 차수만 고려하지만, IPEG 알고리즘은 최소 차수를 가진 체크노드의 ACE 값이 가장 큰 체크노드를 선택하게 된다. ACE를 고려한 IPEG 알고리즘은 PEG 알고리즘 보다 cycle간의 연결성이 높아지고 이는 곧 성능향상으로 이어지게 될 것을 확인 할 수 있다.

나. 순서도

위에서 설명한 내용을 요약하면 IPEG 알고리즘은 마지막 길이의 체크노드 가운데 가장 작은 차수의 체크노드 중 가장 큰 ACE를 가지고있는 체크노드를 선택하는 알고리즘이다. 위에서 설명한 내용을 토대로 본 논문에서 구현한 IPEG 알고리즘을 순서도로 간략하게 표현하면 아래 그림2 와 같다.



< 그림 2. IPEG 알고리즘 순서도 >

4. 모의실험

가. 실험 환경

실험은 BPSK AWGN채널을 통해 신호를 전송하고, SPA를 이용한 복호 과정을 거치면서 BER을 측정하는 방식을 사용하였으며, 부호길이와 부호율이 성능에 어떠한 영향을 끼치는지 알아보기 위해 2가지 부호길이(n= 504, 1008)에 대해서 각각 3가지의 부호율($r_c = 0.5, 0.625, 0.75$) 변화를 주면서 값을 측정했다. 모든 경우에 대해 PEG 알고리즘과 IPEG 알고리즘을 적용하였다. 페리티검사 행렬을 만드는데 사용한 차수 분배는 아래와 같다.

① $r_c = 0.5, (n = 504, 1008)$

$$\lambda(x) = 0.48983x^2 + 0.27384x^3 + 0.108357x^5 + 0.035401x^6 + 0.092563x^{15}$$

② $r_c = 0.625 (n = 504, 1008)$

$$\lambda(x) = 0.40484x^2 + 0.32248x^3 + 0.10059x^5 + 0.05399x^6 + 0.00503x^7 + 0.117597x^{15}$$

③ $r_c = 0.75 (n = 504, 1008)$

$$\lambda(x) = 0.4564x^2 + 0.3251x^3 + 0.1249x^6 + 0.00012x^7 + 0.003674x^{11} + 0.0855x^{12} + 0.00213x^{13}$$

※ 여기서 $\lambda(1) = 1$

알고리즘의 특성상 매년 새로운 페리티체크행렬이 생성 된다. 부호의 평균적인 성능을 알아보기 위해 8~10회 정도 반복 실험한 값의 평균값을 사용하였다.

나. 실험 과정 및 결과

1. PEG알고리즘과 IPEG 알고리즘 비교

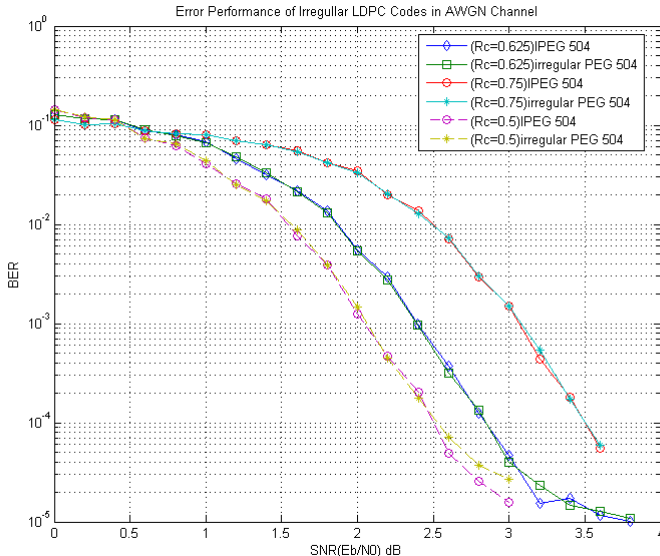
LDPC 부호의 문제점인 높은 SNR에서 오류마루가 발생한다는 점이 PEG알고리즘으로 생성한 부호에서도 발생하였다. IPEG 알고리즘을 통하여 높은 SNR에서 PEG알고리즘에서 발생하는 오류마루 현상을 개선했다. 우리는 여기서 부호율의 변화에 따른 PEG와 IPEG의 성능차이를 비교해보기로 하였다. 그 결과 <표 2>, <표 3>에서 보듯이 부호율이 커질수록 PEG 알고리즘으로 만든 부호와 IPEG 알고리즘으로 만든 부호의 성능차이가 줄어들어 $r_c = 0.625$ 에서는 미비한 성능개선 효과만 있고, 부호율 $r_c = 0.75$ 의 그래프에서는 PEG알고리즘의 BER 곡선과 IPEG알고리즘의 BER 곡선이 거의 동일한 형태를 나타내는 것을 확인할 수 있다.

2. 부호길이의 BER 비교

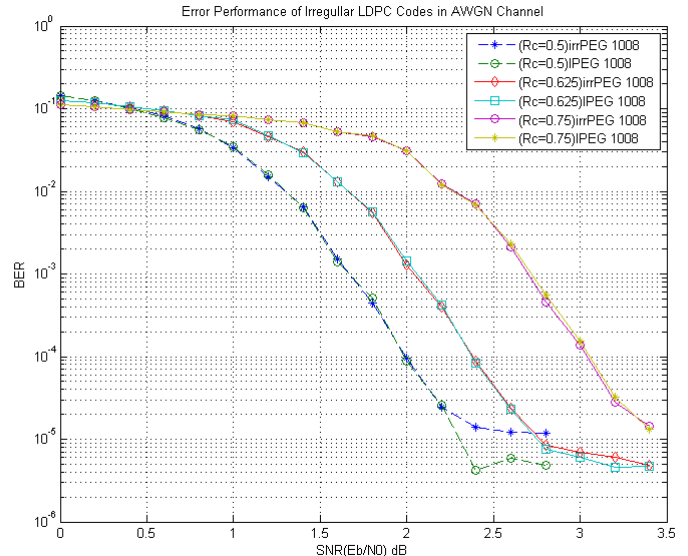
부호길이의 차이에 따른 BER을 비교해보기로 하였다. 부호길이에 따른 성능차이는 IPEG나 PEG나 매우 유사하기 때문에 <표 1>에 IPEG의 경우로만 표현하였다. 부호길이 n이 504에서 1008로 증가하였을 때는 Low SNR에서는 BER이 비슷한 형태로 감소하다가 분기점을 시작으로 성능차이가 발생함을 확인할 수 있다.

3. 부호율에 따른 성능 비교.

PEG와 IPEG 각각의 경우 부호율을 달리하여 성능을 비교해 보았다. 결과는 <표 2>를 통하여 나타내었다. 주목할 점은 PEG보다 IPEG가 성능이 나빠지는 정도가 더 크다는 점을 표를 통해서 확인할 수 있고, 이는 1번 부호율이 0.75일 때 IPEG와 PEG의 성능 차이는 거의 없다는 결과를 다시 한번 설명해준다.



< 그림 3. $n=504$ $r_c = 0.5, 0.625, 0.75$ BER 그래프 >



< 그림 4. $n=1008$ $r_c = 0.5, 0.625, 0.75$ BER 그래프 >

< 표1 부호길이에 따른 성능비교 >

부호율(r_c)		0.5	0.625	0.75
분기점	BER	1×10^{-1}	3×10^{-2}	3.5×10^{-2}
	SNR(dB)	0.0dB	1.35dB	1.9dB
504부호와 1008부호 최대성능차이(dB)		0.8dB	0.75dB	0.45db

< 표2 부호율에 따른 성능 비교 >

n		504	1008	
반전 영역 (BER)	PEG	$< 9.5 \times 10^{-2}$	$< 1 \times 10^{-1}$	
	IPEG	$< 1 \times 10^{-1}$	$< 1 \times 10^{-1}$	
최대 성능 차이 (dB)	PEG	r_c 0.5 \rightarrow 0.625	0.3dB	0.4dB
		r_c 0.5 \rightarrow 0.75	0.9dB	0.9dB
	IPEG	r_c 0.5 \rightarrow 0.625	0.4dB	0.4dB
		r_c 0.5 \rightarrow 0.75	1dB	1.1dB

5. 분석 및 결론

위의 실험 결과를 통해서 IPEG와 PEG 알고리즘 모두 부호길이 증가할수록 부호율이 감소할수록 LDPC 부호의 성능이 향상됨을 확인했다. 부호의 길이가 길어질수록 성능은 만족 할 만큼 향상되지만, 부호 연산의 복잡성도 함께 증가하기 때문에 적정선의 부호길이를 찾는 것이 중요하다. 부호율이 높아질수록 감소하는 BER 성능은 오류를 검색, 정정할 수 있는 패리티 비트의 수가 감소하기 때문이다. 반면에 부호율이 높아질수록 데이터전송률 역시 높아지게 되는데, 여기서 데이터전송률과 BER는 서로 trade-off 관계라고 할 수 있다. 따라서 상황에 따라 적절한 코드율을 정하여 최적화된 코드를 짜는 것이 중요하다. 그리고 부호율이 증가 할수록 IPEG의 성능은 PEG와 비슷해진다. ($r_c=0.75$ 에서는 거의 동일) 이는 이전에 언급하였듯이 패리티 비트 부호의 감소로 인해 LDPC 부호의 특성이 약해져 BER곡선이 알고리즘의 성능 차이보다

SNR의 절대 크기가 중요하게 되기 때문이라고 생각할 수 있다. 따라서 부호율이 증가함에 따라 PEG와 비교하여 IPEG의 장점은 사라져간다고 결론을 지을 수 있다.

부호율이 증가함에 따라 부호의 성능이 나빠지는 것을 막을 수는 없겠지만 나빠지는 정도를 최소화시킬 수 있는 새로운 알고리즘의 연구가 필요하다.

6. 참고문헌

- [1] Xiao-Yu Hu and Dieter M. Arnold "Regular and Irregular Progressive Edge Growth Tanner Graphs," IEEE Trans. inform. theory, vol.51 pp 386-398, Jan. 2005.
- [2] Hua. Xiao and A. H. Banihashemi "Improved Progressive Edge Growth(PEG) Construction of Irregular LDPC codes," IEEE Comm. Letter, vol.8 pp 715-717, Dec. 2004.
- [3] Tao Tian, Chris Jones, John D. Villasenor, and Richard D. Wesel "Construction of Irregular LDPC Codes with Low Error Floors" ULCA, CA, 90095, USA
- [4] <http://ipgdemos.epfl.ch/ldpcopt/> Urbanke's website for calculating Degree Distribution of symbol nodes and check nodes.