

경로검색과 우선경로 기법에 따른
변형된 LDPC 부호의 Backtracking 사후처리 복호기법

양필웅, 박호성, 전보환, 노종선, 신동준*

서울대학교, *한양대학교

{yangpw, lovepark98, netjic}@ccl.snu.ac.kr, jsno@snu.ac.kr, *djshin@hanyang.ac.kr

Modified Backtracking Algorithm of Post-Processing Scheme for LDPC codes
by Path Search and Path Priority

Pilwoong Yang, Hosung Park, Bohwan Jun, Jong-Seon No, and Dong-Joon Shin*
Seoul National Univ., *Hanyang Univ.

요약

본 논문에서는 기존에 소개된 LDPC 부호의 사후처리 복호 기법인 Backtracking 기법의 단점을 보완하는 변형된 사후처리 복호기법을 제안한다. 제안된 부호는 경로검색과 경로에 우선순위를 부여하는 방안을 통해 기존의 사후처리 기법보다 향상된 성능을 보임을 모의 실험을 통해 검증한다.

I. 서론

LDPC (low-density parity-check) 부호는 채널 용량에 근접하는 성능으로 인해 지난 수 년간 많은 주목을 받아 왔다. LDPC 블록 부호의 복호 방식으로 널리 이용되는 BP(belief propagation) 복호방식은 병렬 복호화를 가능케 하면서도 간단한 신뢰 전파 방식으로 복호를 수행함으로써 한계 성능에 근접한 성능을 보임으로 인해 다양한 분야에 적용 되어 왔다.

허나, 이러한 복호 방식을 사용하는 LDPC 부호는 신호 대 잡음비가 증가함에 따라 성능 증가 폭이 급격하게 줄어드는 현상을 보이며, 이를 오류 마루 현상이라 한다. 이러한 오류 마루 현상을 해결하기 위해 현재까지 다양한 접근 방식이 시도 되어 왔으며, 대표적인 접근 방식 중 하나로 사후처리 복호기법의 적용을 고려할 수 있다. [1] 본 논문에서는 사후처리 복호기법을 적용함으로써 오류 마루에서의 성능 한계를 극복 하고자 한다.

본 논문의 구성은 다음과 같다. II장과 III장에서는 각각 기존에 알려진 LDPC 부호의 사후처리 복호기법을 소개하고, 이를 향상 시키기 위한 변형된 사후처리 복호 기법의 적용 방법을 소개한다. IV장 에서 제안된 복호기법의 성능을 살펴본 후 마지막으로 결론을 맺는다.

II. Backtracking 사후처리 복호 기법

[1]에서 소개된 사후처리 복호 기법은 LDPC 부호의 BP 복호 방식을 수행하며, 문제가 되는 변수 노드에의 값에 변형을 가한 후 복호에 성공할 때까지 처음으로 되돌아가 BP 복호를 수행하는 사후처리 방식을 따른다. 이러한 사후처리 방식은 복호기 자체는 그대로 사용하고 변수 노드 값에만 변형이 가해지므로 큰 변화 없이

사후처리 기법을 이용할 수 있다는 장점을 가진다. 기존의 backtracking 사후처리 복호 기법은 아래와 같은 순서를 따른다.

1. 모든 변수노드를 채널값으로 초기화
2. BP 복호기를 수행. 매 번째 반복 때 마다 불만족 하는 체크노드의 개수와 번호를 기록 (C)
3. 복호에 성공했으면 종료, 복호에 실패 했다면 계속해서 4 번의 사후처리 복호 수행
4. 매 번째 반복 때 마다 기록하였던 체크노드의 개수 중 가장 적은 수의 체크노드를 기록했던 반복 때로 돌아가, 해당 체크노드들(C₀)에 연결된 변수노드들(N(C₀))을 후보군으로 기록
5. 후보군 N(C₀) 에 포함된 변수노드 $v \in N(C_0)$ 를 하나 선택하고, 해당 v 변수노드의 LLR (log-likelihood ratio)값(y_v^{LLR})을 아래와 같이 변경

$$y_v^{LLR} = -x(u_{0,v})\eta$$

$x(u_{0,v})$ 는 변수노드 v 의 초기 LLR 값의 부호이며, η 는 사용하는 BP 복호기의 최대 LLR 값이다.

6. 변형된 LLR 값을 가진 변수노드들을 이용하여 처음으로 돌아가 BP 복호기를 수행
7. 복호에 성공했으면 종료. 복호에 실패 했다면 변형했던 변수노드를 초기화 하고, 5 에서 선택한 변수노드 v 를 제외한 다른 변수노드를 선택한 후 재복호를 수행

위와 같은 backtracking 사후처리 복호 기법은 반복 복호가 수행되는 동안 생기는 정보를 이용하여 가장 적은 수의 불만족한 체크노드 수를 보이는 반복 시점에 초점을 맞춘다. 이러한 변형의 근거는 LDPC 부호의 오류 마루 현상에 커다란 영향을 미치는 작은 크기의

트래핑 집합(trapping set)이 이 시점에서 검출할 수 있는 확률이 높다는 이유이다. 트래핑 집합에 포함된 불만족한 체크노드들은 각각 적어도 하나의 트래핑 집합에 포함된 변수노드와 연결되어 있으므로 이 시점에서 불만족한 체크노드와 연결되어 있는 모든 변수노드를 후보군으로 지정함으로써 적어도 불만족한 체크노드 수와 동일한 수의 트래핑 집합에 포함된 변수노드를 후보군에 얻는다. 트래핑 집합을 깨기 위해서는 잘못된 방향으로 변형된 변수노드의 값을 반대값으로 강력하게 변형해줄 필요가 있으므로 BP 복호기에서 허용하는 가장 큰 반대값으로 변수노드 초기값을 설정하게 된다. 이렇게 얻어진 모든 변수노드들에 대해 시행착오법을 수행함으로써 성공적인 사후처리 복호를 진행할 수 있다.

III. 제안된 변형 Backtracking 사후처리 복호 기법

II장에서 소개된 사후처리 복호 기법은 문제가 되는 변수노드를 찾아내고 복호기의 변형 없이 변수노드 초기값 변경만으로 적용할 수 있다는 장점에 비해, 몇 가지 단점을 가진다. 먼저, 최소 개수의 불만족한 체크노드의 수가 일정 개수 이상일 시에는 후보군에 포함되는 변수노드의 수가 너무 많아질 수 있고, 더욱이 높은 부호율을 갖는 부호에서는 하나의 체크노드에 연결된 변수노드의 수가 수십개에 달하므로 시행착오법에 의한 수행이 비효율적일 수 있다.

이러한 문제점을 해결하기 위해 적용할 수 있는 두 가지 기법을 제안한다. 경로검색(path search) 기법과 우선경로(path priority) 기법이며, 이를 적용하여 변경된 기법은 아래와 같다.

1-4. II장과 동일한 수행

- A. 후보군 $N(C_0)$ 를 형성한 체크노드 집합 C_0 에 포함된 체크노드 $c_1, c_2 \in C_0$ 간에 가장 짧은 길이의 경로를 검색한다. 체크노드 c_1 에서 시작하여, 트리전개(tree expectation)에 의해 체크노드 c_2 까지의 경로를 찾을 수 있다.
- B. 검색된 가장 짧은 경로에 포함된 변수노드들의 초기 LLR 값으로 경로에 점수를 지정할 수 있다.

$$\text{score of } p_i = \sum_k |y_0^{LLR}|$$

k 는 경로에 포함된 모든 변수노드를 뜻한다.

- C. 체크노드 집합 C_0 에 포함된 체크노드들 간에 가장 짧은 경로 검색과 이에 지정된 점수에 따라 우선경로를 지정할 수 있다.
- D. 우선 경로에 포함된 후보군의 변수노드들을 바탕으로 후보군에서 먼저 선택될 변수노드의 순서를 지정한다. 이를 바탕으로 backtracking 사후처리 기법을 수행한다.

5-7. II장과 동일한 수행

제안된 사후처리 복호 기법은 기존의 복호 기법에서 후보군 $N(C_0)$ 에 포함되는 변수노드들이 무작위로 선택되어 재복호가 수행된다는 단점을 보완한다. 후보군에 포함되는 변수노드 중 트래핑 집합에 포함되는 변수노드들은 하나의 트래핑 집합을 형성하고 있는 불만족한 체크노드들과 연결되어있고, 오류마루현상을 지배하는 트래핑 집합의 크기가 대부분 알려진 작은 크기임을 고려하여, 하나의 불만족한 체크노드에서 다른

하나의 체크노드로 진행되는 경로가 상대적으로 짧고, 또한 변수노드들의 초기 값이 작은, 즉 신뢰성이 낮은 경로를 우선순위로 둔다. 이렇게 검색한 변수노드들을 우선적으로 사후처리 복호에 시도함으로써 무작위 시도에 비해 적은 반복 횟수를 달성할 수 있다.

V. 모의 실험 결과

모의 실험으로 제안된 사후처리 복호 기법이 기존의 backtracking 기법에 비해 어떠한 장점을 갖는지 확인한다. 사용된 부호는 부호율에 따른 장점을 확인하기 위하여 표준 802.16e LDPC 부호의 $N = 2304$ 길이의 $R = 1/2, 3/4$ 두 가지 부호를 사용하였고, 또한 매우 높은 부호율을 갖도록 설계한 $R = 9/10$ 의 $N = 4552$ QC LDPC 부호를 이용하였다. 오류마루 현상을 보이는 충분히 높은 신호대잡음비 영역에서 오류를 검출하였고, 검출된 오류들을 각각의 사후처리 기법이 처리하는데 필요한 평균 복호 횟수를 확인하였다.

Avg.iter	802.16e (R=1/2)	802.16e (R=3/4)	QC LDPC (R=9/10)
Prev.	111.25	205.71	377.83
Proposed	70.88	144.99	144.18

수 천개 이상의 검출된 오류를 바탕으로 재복호에 성공하기 까지 필요한 복호 횟수를 기록하였으며, 한번의 복호에 수행되는 복호 횟수는 50 회이다. 제안된 사후처리 복호기법은 이전의 복호기법에 비해 재시도 횟수가 적어도 수 회 이상 적어짐을 확인하였다.

VI. 결론

본 논문에서는 기존에 소개된 LDPC 부호의 사후처리 기법인 backtracking 기법의 단점을 보완하기 위한 변형된 사후처리 기법을 제안하였다. 제안된 기법은 불만족한 체크노드 사이의 가장 짧은 경로를 검색하고 이러한 경로에 우선순위를 부과함으로써 무작위로 시도되는 기존 방식의 단점을 해소할 수 있다. 이에 따라 제안된 사후처리 복호기법의 복잡도가 반복횟수 감소에 있어서 장점을 가질 수 있음을 모의실험을 통해 검증하였다.

참고 문헌

- [1] J. Kang, Q. Huang, S. Lin, and K. A.-Ghaffar, "An iterative decoding algorithm with backtracking to lower the error-floors of LDPC codes," IEEE. Trans. on Comm., vol. 59, no. 1, pp. 64-73, Jan. 2011.
- [2] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 599-618, Feb. 2001.
- [3] S. Lin and D. J. Costello, Jr., Error Control Coding, 2nd Ed. NJ: Prentice Hall, 2004.