대한산업공학회/한국경영과학회 2000 춘계공동학술대회 논문집. 2000년4월21-22일. 경남대학교
Session PA5.4

SDP기법에 근거한 전체 네트워크 신뢰도 계산을 위한 효율적 알고리즘
An Algorithm for Computing of the Network Reliability

하경재, 서쌍희
경남대학교 정보통신공학부

Abstract

The network reliability is to be computed in terms of the terminal reliability. The computation of a termial reliability is started with a Boolean sum of products expression corresponding to simple paths of the pair of nodes. This expression is then transformed into another equivalent expression to be a Disjoint Sum of Products form. But this computaion of the terminal reliability obviously does not consider the communication between any other nodes but for the source and the sink. In this paper, we derive the overall network reliability which is the probability of communication that each node in the network communicates with all other remaining nodes. For this, we propose a method to make the SOP disjoint for deriving the netwok reliability expression from the system success expression using the modified Sheinman's method and modified BDD method.

1. 서론

시스템의 신뢰도 계산은 신뢰도 연구의 기본이며, 여러 부품이 네트워크 형태로 구성된 시스템이 고장 없이 성공적으로 작동할 수 있는 확률을 계산하는 것이 신뢰도 계산이며 이는 신뢰도 연구에서 기본이 된다. 이 시스템에 대한 신뢰도 계산의 한 방법으로서, 시스템의 구성을 하나의 도표, 경로리스트 혹은 컷 리스트로 표현하여 이들로부터 논리함수를 얻어내고 그 논리 함수로부터 확률 식을 유도해 낸다. 이 확률 식에 부품이나 링크의 신뢰도 값을 대입하여 시스템의 신뢰도를 계산할 수 있다.

시스템을 논리그래프로서 표현하는 경우, 이로부터 터미널 신뢰도를 구하는 과정은 일반적으로 다음의 두 단계를 거친다.

1. 모든 최소경로(minimal paths)나 컷세트를 구하여 시스템의 성공 혹은 실패함수를 구한다.
2. 시스템의 성공/실패 확률 함수로부터 확률이론, 부울대수, 그래프이론 등을 사용하여 신뢰도 표현식을 유도해 낸다.

첫 번째 단계와 관련하여, 본 논문에서는 네트워크 내에 존재하는 모든 노드쌍들 간의 최소경로를 구하기 위해 신장나무(spanning tree)를 사용한다. 단계 2와 관련하여, 우리는 단계 1에서 구한 시스템 성공함수로부터 전체 네트워크의 신뢰도 표현식을 구하는 알고리즘을 제안한다. 그 중 한 알고리즘은 기존의 논리회로의 합성 및 분석에 많이 사용되어 왔던 BDD(Binary Decision Diagram)[2]를 응용한 것으로서, 주어진 곱의 합(SOP; Sum Of Products) 형태의 시스템 성공함수를 모든 곱항이 서로 배타적이 되도록 만드는 알고리즘이며 다른 하나는 부울 함수를 간단화시키는 방법 중의 하나인 Sheinman[3]의 알고리즘을 변형한 것이다. 결국 이 알고리즘들을 통해 만들어진 형태의 표현식은 배타적 곱의 합(SDP; Sum of Disjoint Products)의 형태가 되며 이것이 신뢰도 표현식이 된다.

따라서 본 논문은 전체 네트워크의 신뢰도 표현식을 효율적으로 구하기 위한 또 다른 방법으로서 BDD를 이용한 알고리즘과 Sheinman을 수정한 알고리즘을 제안하고 이를 다양한 벤치마크 네트워크에 통해 적용한 다음, 기존의 알고리즘과 비교하고자 한다.

2 시스템성공함수를 위한 경로생성알

고리즘

일반적인 컴퓨터 통신 네트워크에서 네트워크 신뢰도를 계산하기 위해서는 얻어진 모든 신장나무들을 부울 대수식의 형태로 표현한 다음 이 부울 대수식을 확률식으로 변환하여 신뢰도를 계산하게 된다.

신장나무(spanning tree)를 구하는 방법은 다음과 같다[15]. 우선, 해당 네트워크에 대한 버텍스컷세트(vertex cutset)를 구한다. 이렇게 얻어진 버텍스 컷세트(vertex cutset) C_i들에 대해 카디시안 곱(Cartesian products) 연산을 적용한다. C_i의 요소들은 그래프 G의 (n-1)개의 모든 노드들에 연결되는 가지(branch)들이다. 이 때 n은 노드의 개수를 의미한다. 그리하여

$$C = C_1 \times C_2 \times \dots \times C_{n-1} = \times_{i=1}^{n-1} C_i$$

여기에서 C 는 (n-1)개의 가지를 가진 G의 서브그래프의 집합이다. (n-1)개의 가지(branch)를 가진 어떤 그래프 G는 C 내에서 짝수 개의 동일한 항을 가질 수 있다[3]. 그러나 이러한 항들은 C로부터 제거되어지고 카디시안곱(Cartesian product) C^*는 짝수 번 반복되지 않는 서브그래프를 포함한다. 따라서, 신장나무(spanning tree)의 개념으로부터 C^* 는 연결 그래프 G의 모든 T_i의 집합이 되게 된다. 전체 네트워크 신뢰도를 계산하기 위하여 이렇게 구해진 신장나무들로부터 시스템 성공함수를 유도할 수 있으며, 시스템 성공함수는 모든 가지(branch)들이 동작 가능한 적어도 하나의 신장나무(spanning tree)가 존재할 확률로 정의할 수 있다.

$$S = T_0 \cup T_i \cup \dots \cup T_{n-1}$$

3. SDP형의 신뢰도 식을 구하는 알고리즘

3.1 기존의 대수적 방법

coherent system의 신뢰도 표현식을 대수적으로 구하는 대표적인 방법으로서 Abraham의 SDP알고리즘[4]을 들 수 있다. 이 알고리즘의 입력데이터는 앞 절에서 구한 최소경로집합이며, 각 알고리즘의 각 수행단계에서 생성된 배타적 항들은 이전 단계에서 이미 구한 모든 항들에 대하여 상호 배타적인 관계를 가진다.

결국, 모든 항들이 배타적으로 되기 때문에 최종 얻어진 부울 표현식은 SDP형이 되며, 이는 각 항의 확률적 합이 터미널 신뢰도가 된다.

이 알고리즘은 다음과 같다[4]. P_i는 어떤 한 임의의 곱항이고 S_j는 단지 보수화 되지 않은 변수를 가지는 한 곱항이라 하자.

단계 1) 만약 S_j 에서는 보수화 되지 않은 형태로 그리고 P_i 에서는 보수형태로 존재하는 적어도 하나의 변수가 있다면 S_j와 P_i는 배타적이다.

단계 2) 만약 S_j와 P_i가 서로 배타적이지 않은 경우, S_j 에서는 보수화 되지 않은 형태로 존재하고 P_i 에서는 존재하지 않는 변수들이 있다면 이들의 집합을 $X=\{X_a, X_b, \dots, X_c\}$라 가정하자. 이 경우

a) 만약 $X = \varnothing$ 이면 $S_j \cup P_i = S_j$ (P_i 내의 항들이 S_j 내에 포함됨)

b)만약$X \neq \varnothing$이면

$$S_j \cup P_i = S_j \cup x'_a P_i \cup x_a x'_b P_i \cup \dots \cup X_a X_b \dots X_c P$$

3.2 SDP 표현식을 구하는 제안 알고리즘-1

본 논문에서 제안하는 알고리즘은 Sheinman 방법[5]을 기본으로 하고 있기 때문에 먼저 k 쌍의 터미널 간 최소 경로들에 대한 부울 합의 식을 정형(canonical form)으로 만들고 각 최소항(minterm)을 해당 10진수 값으로 바꾼다. 그리고 해당 터미널 쌍에 관계하는 최소 항들에 표기를 한다. 이 알고리즘은 Shannon의 전개이론에 근거하는데, 정형의 부울함수를 다음과 같은 형태로 변수 xi 에 대하여 다음과 같이 반복 전개하는 방법이다.

$$f = x_i \cdot f|_{xi=1} + x_i' \cdot f|_{xi=0}$$

Sheinman 알고리즘[5]을 우리의 문제에 맞게 다음과 같이 변경 제안한다.

단계 1) 주어진 각 터미널 쌍에 대한 성공함수의 곱항 모두의 해당 십진수를 크기대로 열 방향으로 배치시킨다.

단계 2) 이 십진수들을 두 그룹으로 나누되 한 그룹은 x_1', 또 다른 하나는 x_1 그룹으로 한다. 여기서 x_1' 그룹에 포함되는 십진 숫자들은2^{n-1}(x1의 이진 가중치값)보다 작은 것들이고, x_1 그룹에 포함되는 십진 숫자들은 2^{n-1}보다 같거나 큰 숫자들

이다. 특히 x1 그룹에 속하는 십진 숫자에 2^{n-1}을 빼 준 숫자를 크기대로 나열한다.

단계 3) 단계 2)에서 나열한 십진 숫자들 중에서 같은 숫자들(x_1'그룹과 x_1그룹에 공통으로 속해 있는 숫자)은 다시 세 번째 열에 -그룹으로 포함시키고 x_1' 그룹과 x_1 그룹의 해당 숫자들은 마크해둔다. 이 마크의 의미는 이 들 숫자들은 열 x_1'그룹과 x_1그룹에 중복되어 있다는 것이다.

단계 4) 각 그룹 열을 검사하여 해당 그룹에 속하는 십진수가 모두 마크되어 있으면 그 그룹에 대해서는 다음 단계로 더 이상 전개하지 않는다. 그렇지 않고 만일 마크되어 있지 않은 십진수가 하나 이상 있으면 그 숫자들만에 대하여 다시 단계 3)과 같이 전개한다. 이 과정을 변수 x_n의 경우까지 반복 전개한다.

단계 5) 최하위 변수 x_n 까지 전개되는 경우는 그 나머지는 0이 되도록 한다.

단계 6) 0으로 까지 전개된 각 열에 대하여 거꾸로 처음까지 추적하면 서로 배타적이 되는 모든 곱 항들을 구할 수 있다.

이와 같은 알고리즘을 사용하면 여러 개의 터미널 쌍에 대해서도 단계 1)에서 모든 최소항들을 열거하여 어떤 터미널 쌍에 해당하는 최소항인지를 표시만 해 두면 한 번의 전개로 모든 터미널 쌍에 대한 신뢰도식을 구할 수 있다.

3.3 SDP 표현식을 구하는 제안 알고리즘-2

함수 f는 |G| 노드를 갖는 축소된 함수 그래프로 볼 수 있으며 이 그래프가 부울 함수를 표현하는 기본 데이터 구조가 된다. 특히 본 알고리즘은 출력 1을 나타내는 입력변수집합을 구하기 위해 그래프를 역추적하는 작업을 수행하지 않는다. 따라서, 알고리즘이 보다 간단하며 수행시간면에서도 효과를 볼 수 있다.

이 알고리즘은 다음과 같다.

단계 1) 우선, 함수 f_0를 구하기 위하여 x_1=0으로 정한다. 또한 f_1을구하기 위하여 x_1=1로 정한 다음 각각 여인자함수 f_0와 f_1을 구한다.

단계 2) 변수 x_2에 대해서도 동일한 방식으로 전개가 이루어진다. 만약 두 개의 가지가 같은 함수를 가리키게 되면 그 가지들은 같은 노드를 가리키도록 만든다.

단계 3) 모든 경로들이 0 혹은 1로 끝날 때까지 전개가 계속된다.

단계 4) 함수값이 1이 되는 여인자함수의 입력변수 값 집합 찾는다.

4. 적용결과 및 분석

본 논문에서는 7개의 벤치마크 네트워크에 각 알고리즘을 적용해 보았다. 그 결과는 표 1과 같다. 표 1로부터 우리는 다음과 같은 결론을 추론할 수 있다. 네트워크의 크기가 커지게 되면 생성되는 시스템성공함수내의 곱 항의 수는 기하급수적으로 늘어나게 된다. 이는 Abraham방법의 경우 상대적으로 비교해야 할 항의 개수가 크게 늘어나게 된다는 것을 의미한다. 따라서 그 수행시간이 현저히 떨어지게 된다. 이에 반해 제안 알고리즘-1의 경우 시스템성공함수내의 곱항의 수와 관계없이 데이터를 처리함으로써 네트워크의 크기가 커진다 하더라도 단순히 처리해야 할 십진수의 개수가 증가하는 것이므로 Abraham의 방법에 비해 수행 시간면에서 보다 좋은 결과를 나타낼 수 있다. 제안 알고리즘-2 방법의 경우 시스템성공함수내에 존재하는 각 입력변수 개수만큼의 확장이 이루어지며 각 확장단계에서 각 입력변수들에 대한 값들이 저장되게 된다. 따라서 생성되는 SDP곱항들은 단순히 저장된 값들을 출력하기만 하면 된다. 따라서 네트워크의 크기가 커질 경우 역시 수행 시간면에서 보다 좋은 효과를 나타낼 수 있다.

네트워크	평가 기준	Abraham	수정된 Sheinman	수정된 BDD
네트워크 1	생성된 SDP항의 수	8	8	8
	Computing time	0.001	0.006	0.005
네트워크 2	생성된 SDP항의 수	10	12	12
	Computing time	0.004	0.018	0.015
네트워크 3	생성된 SDP항의 수	16	18	16
	Computing time	0.11	0.15	0.125
네트워크 4	생성된 SDP항의 수	36	35	33
	Computing time	1.525	1.348	1.326
네트워크 5	생성된 SDP항의 수	77	74	76
	Computing time	3.282	3.039	3.046
네트워크 6	생성된 SDP항의 수	190	173	175
	Computing time	16.185	13.230	12.220
네트워크 7	생성된 SDP항의 수	705	684	672
	Computing time	60.590	45.924	45.323

표 1. 전체 신뢰도

결론적으로, 전체 네트워크 신뢰도 계산 시

제안된 알고리즘들은 네트워크의 크기가 작은 경우 Abraham의 방법에 비해 계산시간과 평균 생성된 SDP항의 수면에서 거의 동일한 반면 네트워크의 크기가 커질수록 보다 좋은 효과를 나타내었다.

5. 결론

본 논문은 일반적인 네트워크의 전체 네트워크 신뢰도를 구하는 효율적인 알고리즘을 제안하였다. 이 방법은 모든 노드 쌍들간의 모든 최소경로집합을 신장나무를 이용해 성공함수를 구한 후 이 함수식을 SDP 표현식으로 구하는 대수적인 방법을 기본으로 한다[9,10]. 제안된 첫 번째 알고리즘은 신뢰도 계산을 위한 신뢰도식 유도시 각 곱항을 상호배타적으로 만드는 알고리즘으로서 부울함수 간단화를 위해 사용되는 Sheinman의 알고리즘을 변형한 것이며 두 번째 알고리즘은 표준 BDD 알고리즘을 변형한 것이다.

제안된 알고리즘들은 기존의 알고리즘에 비해 알고리즘이 단순하며, 컴퓨터 상에서 구현이 용이하고, 대규모의 네트워크에 보다 효율적인 것으로 나타났다. 이를 위해 본 논문에서는 7개의 벤치마크 네트워크[8]에 이 알고리즘들과 기존의 Abraham의 방법을 적용시켜 비교, 평가해 보았다. 그 결과, 시스템성공함수내의 항수에 관계없이 입력변수의 수에 따라 확장이 이루어졌으며 이는 네트워크의 크기가 커질 경우 기존의 방식에 비해 보다 빠른 처리가 가능하였다.

REFERENCE

[1] K. K. Aggarwal, *Reliability Engineering*, Kluwer Academic Pub. 1993

[2] Sheldon B. Akers, "Binary Decision Diagram", IEEE Trans. On Computers, Vol. c-27, pp. 509-516 1978

[3] M. Piekarski, "Listing of all possible trees of a linear graph", IEEE Trans. Circuit Theory, Vol. CT-12, pp. 124-125, Mar. 1965.

[4] J. A. Abraham, "An Improved method for Network Reliability", IEEE Trans. on Reliability, Vol. R-28, pp.58-61, Apr. 1979.

[5] A. H. Scheinman, "A Method for Simplifying Boolean Functions", The Bell System Technical Journal, pp. 1337-1346, 1962.

[4] M. O. Locks, "Recursive disjoint products: a review of three algorithms", IEEE Trans. on Reliability, Vol R-31, pp. 33-35, 1982.

[5] K. D. Heidtmann, "Smaller Sums of Disjoint Products by Subproduct Inversion", IEEE Trans. on reliability, Vol. R-38, pp. 305-311, 1989.

[6] H. A. AboElFotoh, C. J. Colbourn, "Efficient Algorithms for Computing the Reliability of Permutation and Interval Graphs", Networks, Vol. 20, pp. 883-898, 1990.

[7] G. Hartless, L. Leemis, "Computational Algebra Applications in Reliability Theory", IEEE Trans. on Reliability, Vol. 45, No. 3, pp. 393-399, 1996.

[8] S. Soh, S. Rai, "Experimental Results on Preprocessing of Path/Cut Terms in Sum of Disjoint Products Technique", IEEE Trans. on Reliability, Vol. 42, No. 1, pp. 24-33, 1993.

[9] S. Hariri, C. S. Raghavendra, "SYREL: A Symbolic Reliability Algorithm Based on Path and Cutset Methods", IEEE Trans. on Reliability, Vol. 36, No. 10, pp. 1224-1232, 1987.

[10] N. Deo, *Graph Theory with Applications to Engineering and Computer Science*, Prentice Hall Inc. 1974.