대한산업공학회/한국경영과학회 2000 춘계공동학술대회 논문집. 2000년 4월 21-22일. 경남대학교
Session OC6.5

다중 터미널 신뢰도 계산을 위한 효율적 알고리즘
An Algorithm for Computing the Reliability of Multiple Terminal-Pairs

하경재, 김원경
경남대학교 정보통신공학부

Abstract

The computation of a terminal-pair 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 Sum of Disjoint Products form. The terminal reliability can then be obtained in a straightforward manner from the Sum of Disjoint Products form. In this paper, we proposed an efficient method to obtain reliability expressions for calculating multiple K terminal-pairs reliability between terminal-pairs of nodes of computers of complex networks. Actual programming results show that the new method is superior with regard to computational efficiency, especially for computing the reliability of multiple terminals.

1. 서론

터미널 신뢰도는 컴퓨터 네트워크의 고신뢰도 설계에 매우 중요한 요소이다. 터미널신뢰도란 네트워크상의 두 컴퓨팅노드 사이에 정상 경로가 적어도 하나가 존재할 확률로서 정의된다. 따라서, 신뢰도를 나타내는 기호적인 표현식을 구하는 것이 무엇보다 중요하다. 즉, 대부분의 컴퓨터 네트워크는 고정된 토폴로지를 갖는 대신에 그 구성요소들의 신뢰도는 시간에 따라 변화하기 때문에 이 표현식을 이용하면 쉽게 터미널간의 신뢰도를 재 계산해 낼 수 있다. 또한, 주어진 비용 제약 하에서 네트워크의 터미널 신뢰도를 최적화하는데 반드시 필요한 파라메터도 이 신뢰도 표현식을 이용하면 쉽게 얻을 수 있다.[1][2]

신뢰도 표현식을 구하는 방법으로는, 두 터미널사이의 경로나 컷 세트를 이용한 성공함수를 부울 대수적으로 각 곱 항이 서로 배타적이 되도록 하여 구하는 효과적인 방법들이 널리 알려져 있다.[8-9] 이 들 방법에서 제안한 알고리즘은 각 수행단계마다 상호 배타적인 확률적 사건이 되도록 즉, SDP(Sum of Disjoint Poducts)형태로 유도하는 이른바 disjoint process가 필요하며 이를 수행하는 시간이 많이 걸린다는 문제점이 있다[3].

따라서, 본 논문에서는 네트워크 그래프의 두 터미널 간의 경로집합을 이용한 부울 성공함수로부터 SDP형의 신뢰도 표현식을 유도하는 새로운 방법을 제안하고자 한다. 이 방법은 부울 함수의 간략화에 사용되는 Sheinman 방법[5]을 개선한 것이며, 특히 제안된 방법을 사용하면 한번의 수행과정만을 통하여 임의의 K 터미널신뢰도들을 바로 구할 수 있어 대규모의 네트워크에서 다중 터미널 신뢰도를 구하는 데 효과적이다.

2. SDP형의 신뢰도식을 구하는 알고리즘

2-1. 성공함수를 위한 경로계산

신뢰도 표현식을 구하는 데 있어 가장 먼저 요구되는 것은 터미널노드 간의 최소경로(minimal paths)들의 집합이다. 간단한 네트워크인 경우는 직관적으로 구할 수 있지만 네트워크의 규모가 커지면 보다 체계적인 방법이 필요하다. 최소경로집합을 구하는 효율적인 방법이 많이 발표되어 왔지만 [1][6][7], 본 논문에서는 네트워크 그래프를 연결행렬(connection matrix)로 표현하고 이 연결행렬에 대하여 출발지 노드에 해당하는 행(row)과 목적지 노드에 해당하는 열(column)을 삭제한 성공 행렬식을 부울대수 법칙에 따라 계산하는 방법[6]을 그대로 적용한다.

<그림 1>의 네트워크에서 하나의 터미널 쌍 (n_1->n_6) 의 최소 경로집합은 다음과 같이 성공행렬식 $|S|_{16}$을 계산하여 식 (1)과 같이 성공함수를 구한다.

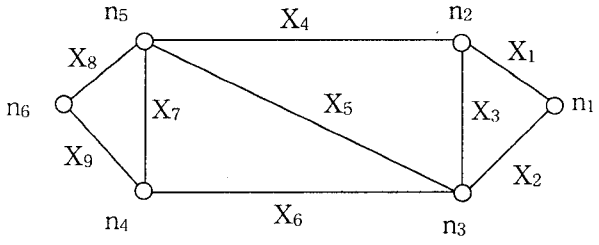

그림 1. 네트워크 예

$$|S|_{10} = \begin{array}{c|cccccc} & n_1 & n_2 & n_3 & n_4 & n_5 & n_6 \\ n_1 & 0 & X_1 & X_2 & 0 & 0 & 0 \\ n_2 & 0 & 0 & X_3 & 0 & X_4 & 0 \\ n_3 & 0 & X_3 & 0 & X_6 & X_5 & 0 \\ n_4 & 0 & 0 & 0 & 0 & X_8 & X_9 \\ n_5 & 0 & 0 & X_5 & X_7 & 0 & X_7 \\ n_6 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$

$$|S|_{16} = X_1X_3X_6X_7X_8 \cup X_1X_3X_6X_9 \cup X_1X_3X_5X_8 \cup X_1X_3X_5X_7X_9 \cup X_1X_4X_8 \cup X_1X_4X_7X_9 \cup X_1X_4X_5X_6X_9 \cup X_2X_6X_7X_8 \cup X_2X_6X_9 \cup X_2X_5X_8 \cup X_2X_5X_7X_9 \cup X_2X_3X_4X_8 \cup X_2X_3X_4X_7X_9 \quad \cdots (1)$$

2-2. 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 disjoint 항들을 만들어 나가게 된다. 이 단계에서 생성된 disjoint항들은 이전단계에서 이미 구한 모든 항들에 대하여 disjoint관계를 가진다.
결국, 모든 항들이 disjoint되기 때문에 최종 얻어진 부울표현식은 SDP형이 되며, 이는 각 항의 확률적 합이 터미널 신뢰도가 된다.
이 알고리즘은 다음과 같다.[4]

S_i를 보수가 아닌 변수들로만 구성된 하나의 부울 곱항이라 하고, P_i 는 임의의 곱항이라 하자.

단계 1) P_i 에 있는 보수형태의 변수들이 S_j 내에 적어도 하나 이상이 있으면 P_i와 S_j 는 서로 disjoint하다.

단계 2) S_j 와 P_i 가 disjoint하지 않으면, $X=\{ X_a, X_b, ...,X_k\}$를 S_j 에는 있고 P_i 에는 없는 변수들의 집합이라 하자. 그러면

(1) 만일 $X=\varnothing$ 이라면, $S_j \cup P_i = S_j$
(2) 만일 $X \neq \varnothing$ 이라면, $S_j \cup P_i = S_j \cup X_a'P_i \cup X_aX_b'P_i \cup ... \cup X_aX_b...X_kP_i$

2-3 다중 SDP 표현식들을 구하는 제안 알고리즘

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 터미널쌍에 관계하는 최소항들에 표기를 한다. 이 알고리즘은 Shannon의 전개이론에 근거하는 데, 정형의 부울함수를 다음과 같은 형태로 변수 x_i 에 대하여 다음과 같이 반복전개 하는 방법이다.

$$f = x_1 \cdot f_i \mid_{x_i=1} + x_1' \cdot f_i \mid_{x_i=0} \quad \text{------(2)}$$

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

단계 1) 주어진 각 터미널 쌍에 대한 성공함수의 곱항 모두의 해당 10진수를 크기대로 열 방향 배치시킨다.

단계 2) 이 10진 수 들을 두 그룹으로 나누되 한 그룹은 x_1', 또 다른 하나는 x_1 그룹으로 한다. 여기서 x_1' 그룹에 포함되는 10진 숫자들은 2^{n-1}(x_1의 이진 weight값)보다 작은 것들이고, x_1 그룹에 포함되는 10진 숫자들은 2^{n-1}보다 같거나 큰 숫자들이다. 특히 x_1 그룹에 속하는 10진 숫자에 2^{n-1}을 빼준 숫자를 크기대로 나열한다.

단계 3) 단계 2)에서 나열한 10진 숫자들 중에서 같은 숫자들(x_1'그룹과 x_1 그룹에 공통으로 속해있는 숫자)은 다시 세 번째 열에 - 그룹으로 포함시키고 x_1' 그룹과 x_1 그룹의 해당 숫자들을 마크해둔다. 이 마크의 의미는 이 들 숫자들은 열 x_1'그룹과 x_1 그룹에서는 중복되어 있다는 것이다.

단계 4) 각 그룹 열을 검사하여 해당 그룹에 속하는 10진수가 모두 마크되어 있으면 그 그룹에 대해서는 다음 단계로 더 이상 전개하지 않는다. 그렇지 않고 만일 마크되어 잇지 않는 10진수가 하나이상 있으면 그 숫자들만에 대하여 다시 단계 3)과 같이 전개한다. 이 과정을 변수 x_n 의 경우까지 반복 전개한다.

단계 5) 최 하위변수 x_n 까지 전개되는 경우는 그 나머지는 0 이 되도록 한다.

단계 6) 0으로 까지 전개된 각 열에 대하여 거꾸로 처음까지 추적하면 서로 disjoint되는 모든 곱항 들을 구할 수 있다.

이와 같은 알고리즘을 사용하면 다중 터미널 쌍에 대해서도 단계 1)에서 모든 최소항들을 열거하여 어떤 터미널 쌍에 해당하는 최소항 인지를 표시만 해두면 한번의 전개로 모든 터미널 쌍에 대한 신뢰도식을 구할 수 있다.

이 알고리즘을 실제 그림 1의 네트워크 성공함수식 (1)에 대하여 다음 식 (3)과 같은 터미널 쌍 n1->n6 의 SDP 식을 구할 수 있다.

SD_{16} = x1x2'x3x4'x5x6'x7x8'x9 +
x1x2'x3x4'x5x6'x7x8x9' +
x1x2'x3x4'x5x6'x8x9 +
x1x2'x3x4'x5x6x7'x8'x9 +
x1x2'x3x4'x5x7'x8x9 + x1x2'x3x4'x6x7x8'x9
+ x1x2'x3x4'x6x7x8x9' + x1x2'x3x4'x6x8x9 +
x 1 x 2 ' x 3 x 4 x 5 ' x 6 ' x 7 x 8 x 9

+x1x2'x3x4x5'x6'x7x8x9' + x1x2'x3x4x5'x6'x8x9 + x1x2'x3x4x5'x7'x8x9' + x1x2'x3x5'x6x7'x8'x9 + x1x2'x4x5x6'x7x8'x9 + x1x2'x4x5x6'x7x8x9' + x1x2'x4x5x6'x8x9 + x1x2'x4x5x6x7'x8'x9 + x1x2'x4x5x7'x8x9' + x1x2'x4x6x7x8'x9 + x1x2'x4x6x7x8x9' + x1x2'x4x6x8x9 + x1x3'x4x5'x6'x7x8'x9 + x1x3'x4x5'x6'x7x8x9' + x1x3'x4x5'x6'x8x9 + x1x3'x4x5'x7'x8x9' + x2x3x4x5'x6'x7x8'x9 + x2x3x4x5'x6'x7x8x9' + x2x3x4x5'x6'x8x9 + x2x3x4x5'x7'x8x9' + x2x5x6'x7x8'x9 + x2x5x6'x7x8x9' + x2x5x6'x8x9 + x2x5x7'x8x9' + x2x6x7x8x9'+ x2x6x9 ---------- (3)

3. 적용결과 및 분석

네트워크그래프의 노드수가 많아질수록 노드들 사이의 경로 수는 기하급수적으로 늘어난다. 따라서 신뢰도를 구하는 방법들을 비교할 경우 복잡도에 의하기보다는 최종 신뢰도표현식에 포함되는 곱항의 수와 표현식을 구하는 데 걸리는 시간 등의 관점에서 비교한다. 본 논문에서 제안한 알고리즘과 기존의 방법을 그림 1에 적용한 결과를 표 1과 표 2에 나타낸다. 표 1은 각각 하나의 터미널 쌍에 대한 신뢰도식을 구할 경우이며 표 2는 k 개의 다중 터미널쌍 들에 대한 경우이다. 단, 링크 x1, ..., x9 의 경로신뢰도를 각각 p1=0.9, p2=0.8, p3=0.7, ... , p9=0.1 로 하였다.

표 1. 단일 터미널 신뢰도

Algorithm	Terminal	Generated number of SDP terms	Computing time(sec.)	Terminal reliability
Abraham	n_1 -> n_6	41	0.015	0.2002306
Our algorithm		35	0.05	0.2002306
Abraham	n_2 -> n_4	17	0.0125	0.5198145
Our algorithm		22	0.05	0.5197145
Abraham	n_3 -> n_4	13	0.0125	0.5459723
Our algorithm		10	0.05	0.5459723

표 1 에서는 본 논문에서의 방법이 생성되는 평균 SDP항의 수는 비슷하지만 모든 경우에 그 계산시간은 오히려 오래 걸린다. 따라서 하나의 터미널 신뢰도를 구할 때는 본 논문의 방법이 좋지 못하다. 그러나 같은 세 터미널 쌍에 대한 신뢰도를 한꺼번에 구하는 경우에는 표 2에서와 같이 기존의 방법은 총 계산시간은 산술적인 시간 합이 되나, 본 논문에서의 방법은 하나의 터미널 쌍에 대한 신뢰도를 구하는 시간보다 약간만 증가할 뿐이다. 따라서 이 표 2 는 신뢰도를 구하고자 하는 터미널 쌍이 아주 많을 경우에는 본 논문에서의 방법이 매우 유리할 것임을 시사하고 있다. 다양한 노드 쌍에 대한 적용결과 노드 쌍의 수와 그에 대한 평균 신뢰도 계산시간의 관계를 그림 2 에 보인다. 이 그림에 의하면 비록 하나의 네트워크에 대해서만 적용한 결과이나, 본 논문에서의 방법이 다중 신뢰도 계산에 유리하다는 것을 알 수 있다.

표 2. 터미널 신뢰도

Algori-thm	Multi-terminal Reliability	Generated number of SDP terms	Comput-ing time (sec.)	Terminal reliability
Abraham	n_1 -> n_6 n_2 -> n_4 n_3 -> n_4	71 (41, 17, 13)	0.025	0.2002306 0.5198145 0.5459723
Our algorithm		67 (35, 22, 10)	0.065	0.2002306 0.5198145 0.5459723

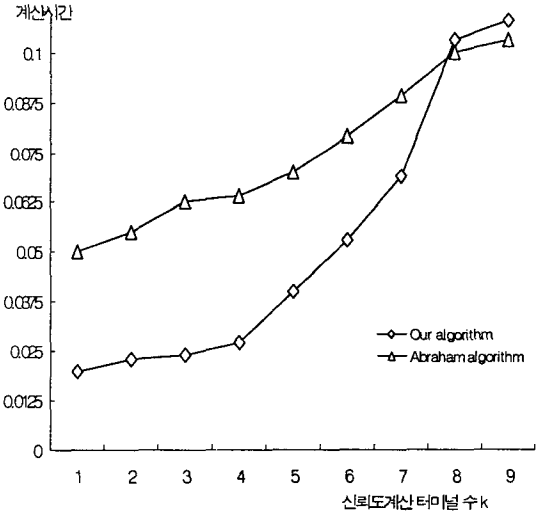

그림 2. 다중 터미널 신뢰도 계산시의 두 알고리즘 비교

4. 결론

본 논문은 일반적인 네트워크의 터미널 신뢰도를 구하는 효율적인 알고리즘을 제안하였다. 이 방법은 주어진 노드 쌍들에 대한N 최소경로집합을 구해, 성공함수를 구한 후 이 함수식을 SDP 표현식으로 구하는 대수적인 방법을 기본으로 한다. 제안 알고리즘은 기존의 부울함수 간단화에 사용하는

Sheinman방법을 변형 적용하여, 주어진 다중 성공 함수들을 SDP형으로 만드는 방법이다. 이 방법의 특징은 구하고자 하는 터미널신뢰도의 수에 관계없이 한번의 수행만으로 k개의 모든 터미널신뢰도를 모두 구할 수 있다는 것이다. 본 논문에서의 방법이 유용하다는 것을 입증하기 위해 기존의 방법과 본 알고리즘을 하나의 네트워크그래프에 적용한 결과, 본 논문에서의 방법이 다중 터미널신뢰도를 구하는 데 계산 시간 면에서 그 효용성이 뛰어나다는 것을 알 수 있었다. 더구나, 간단히 10진 숫자를 대상으로 한 처리를 기본으로 하기 때문에 구현이 쉬울 뿐만 아니라, 네트워크의 규모가 커질수록 그 효용성은 더욱 크리라 사료된다. 앞으로 더욱 다양한 네트워크에 대한 적용결과를 통하여 그 타당성을 보이고 다른 기존의 방법들과도 비교 고찰할 예정이다.

REFERENCES

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

[2] C. S. Raghavendra and S. Hariri, "Reliability optimization in the design of distributed systems," IEEE Trans. Software Eng., vol. SE-11, pp. 1184-1193, Oct. 1985.

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

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

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

[6] S. Rai, K. K. Aggarwal, "An Efficient Methods for Reliability Evaluation of a General Network", IEEE Trans. on Reliability, Vol. R-27, No.3, pp.206-211, Aug. 1978.

[7] K. K. Aggarwal, J. S. Gupta, K. B. Misra, "A Simple Method for Reliability Evaluation of a Communication System", IEEE Trans. on Communication, Vol. COM-23, pp.563-565, May 1975.

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

[9] M.O.Locks, "Recursive Disjoint Products: a Review of Three Algorithms", IEEE Trans. on Reliability, Vol. R-31, No. 1, pp.33-35, Apr. 1982.