2002 Proceedings of the Autumn Conference, Korean Statistical Society

An Algorithm for Joint Reliability Importance in Networks

장규범 1), 박동호 2), 이승민2)

요약

네트워크를 설계하거나 평가하는데 있어 중요한 문제 중 하나는 그 네트워크를 구성하는 요소들간의 상대적 중요도(importance)에 관한 문제이다. 이런 중요도를 나타내는 여러 가지 측도들 중 하나인 Joint Reliability Importance(JRI)는 Hong & Lie(1993)에 의해 소개되었으며, 네 가지 파생된 서브그래프의 신뢰성을 구하여 JRI를 계산하는 방법이 제시되었다. 본 연구에서는 minimal path set을 이용하여 파생되는 서브 그래프 신뢰성 계산에서의 중복되는 계산과정을 줄임으로써 JRI를 보다 효율적으로 구하는 방법을 제시하고자 한다.

주요용어 : Joint Reliability Importance, Sum of disjoint products, Abraham-Locks-Revised Algorithm(ALR Algorithm)

1. 서론

...

통신망, 교통망, 전력망과 같은 네트워크(network) 구조를 갖는 복잡한 시스템의 수학적인 모형으로서 그래프(graph)가 주로 사용되고 있다. 하나의 네트워크는 노드(node)로 이루어진 집합 V와 링크(link)로 이루어진 집합 E로 구성된 그래프 $G(V,E)$로 표현될 수 있다. 이러한 네트워크를 설계하거나 평가하는데 있어 중요하게 다루어야 할 점 중 하나는 그 네트워크를 구성하는 요소들간의 상대적인 중요도(importance)에 관한 문제이다. 중요도는 네트워크의 설계자나 관리자들에게 어떤 구성요소가 네트워크의 신뢰성에 얼마만큼 영향을 주는가, 어떤 구성요소를 우선적으로 향상시킴으로써 전체 네트워크의 신뢰도를 최대로 향상시킬 수 있는지에 관한 정보를 제공한다. 이러한 중요도를 나타내는 여러 가지 측도들 중 하나인 Joint Reliability Importance (JRI)는 네트워크의 신뢰성(reliability)에 대해 두 링크 i와 j가 미치는 영향에 관한 측도로서, Hong & Lie(1993) 논문에 의하여 제시되었다. JRI를 계산하는 문제는 전체 네트워크에서 파생된 네 가지 서브 네트워크의 신뢰성을 계산하는 문제로 귀착된다. 네트워크의 신뢰성을 구하는 문제는 일반적으로 구성요소들의 수가 증가할수록 많은 계산시간과 계산량이 요구되기 때문에 매우 복잡하여 다루기가 힘든 경우가 대부분이다. 본 연구에서는 Locks(1987)의 논문에서 제시된 네트워크의 신뢰성 계산방법(Abraham-Lock-Revised 알고리즘)을 사용하여 네 가지 파생된 서브그래프의 신뢰성을 계산하는 과정에서, 중복되는 계산과정을 최대한 줄임으로써 JRI를 보다 효율적으로 구하는 방법을 제시하고자 한다.

---

1) 한림대학교 통계학과 석사과정
2) 한림대학교 수리정보과학부 교수

2. Joint Reliability Importance

가정

1. 네트워크 내의 노드는 항상 작동한다.
2. 네트워크 내의 링크는 작동하는 상태와 작동하지 않는 상태의 두 가지 상태를 갖는다.
3. 링크의 고장(혹은 작동)은 서로 독립적(independent)이다.

네트워크에서의 두 링크 i와 j에 대한 Joint Reliability Importance는 Hong & Lie(1993) 논문에서 다음과 같이 정의되었다.

$$\mathrm{JRI}\ (i,j) = \frac{d^2R(G)}{dp_i dp_j}$$

여기에서 R(G)는 네트워크의 source node와 terminal node가 작동하는 링크에 의해 연결될 확률이며, $G*i$는 링크 i가 항상 작동하는 경우를, $G-i$는 링크 i가 항상 작동하지 않는 경우를 나타낸다. 이를 통하여 $\mathrm{JRI}(i,j) = R(G*i*j) - R(G*i-j) - R(G-i*j) + R(G-i-j)$로 표현할 수 있으며, 이를 이용하여 네 개의 서브 그래프 $G*i*j$, $G*i-j$, $G-i*j$, $G-i-j$의 신뢰성을 각각 계산함으로써 JRI(i,j)를 얻어낼 수 있다. 본 연구에서는 신뢰성를 계산하는 방법으로서 ALR 알고리즘을 사용하고 있으나 파생되는 네 가지 서브그래프의 minimal paths를 생성하여 신뢰성을 각각 계산하기보다는 네트워크의 minimal paths를 다른 방법으로 분류하여 중복되는 과정을 줄이고자 한다. 네트워크의 두 링크 i와 j에 대하여 주어진 minimal paths를 다음과 같이 A_i, A_j, A_{ij}, A_x를 분류한다.

A_i : 링크 i 만 포함하는 path로 이루어진 집합, 링크 i를 path에서 삭제.

A_j : 링크 j 만 포함하는 path로 이루어진 집합, 링크 j를 path에서 삭제.

A_{ij} : 링크 i와 j를 모두 포함하는 path로 이루어진 집합, 링크 i와 j를 path에서 삭제.

A_x : 링크 i와 j를 모두 포함하지 않는 path로 이루어진 집합.

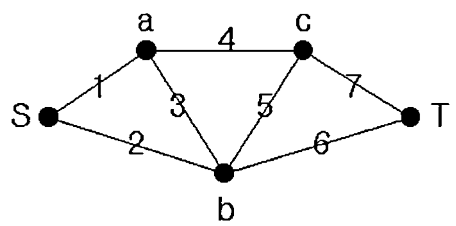
이때, 분류된 A_i, A_j, A_{ij}, A_x들은 minimal path가 아닐 수 있다. 또한, 계산과정에서의 편의상 네 개의 집합은 모두 path의 size와 lexicographical ordering에 의하여 정렬한다. JRI(i,j)를 계산하기 위한 알고리즘은 다음과 같다.

ALGORITHM:

1. (initialize) 링크 i, j를 선택, minimal paths를 A_i, A_j, A_{ij}, A_x로 분류,
   REL = 0, REL1 = 0, REL2 = 0.
2. Set $P = A_x$, $S = A_j$,
   2.1. If $S = \phi$, Go to step 4.
   2.2 Call FUNCTION(P,S), REL1 = REL.
3. Set $P = A_i \cup A_x$, $S = A_j$,
   3.1 Call FUNCTION(P,S), REL2 = REL.
4. Set $P = A_j \cup A_i \cup A_x$, $S = A_{ij}$,
   4.1. If $S = \phi$, Go to step 5.
   4.2 Call FUNCTION(P,S), REL2 = REL2 + REL.
5. JRI(i,j) = REL2 - REL1.

서브프로그램 FUNCTION( )에서 생성되어 저장된 값(REL)은 reliability polynomial에 일대일 대응되는 boolean polynomial 형태이며, FUNCTION( )에서 sum of disjoint products를 생성하기 위하여는 Locks(1987)의 ARL algorithm의 Inner Loop를 사용하는 방법을 채택한다. 본 연구에서는 ARL algorithm을 사용하는데 있어, 두 집합의 paths를 비교하여 disjoint products를 생성하게 된다. 집합 P가 공집합일 경우는 집합 S의 path만으로 disjoint products를 생성하고 공집합이 아닌 경우에는 집합 S의 path를 집합 P의 path와 비교하여 disjoint products를 생성하게 된다. 만약 집합 S의 path가 두 개 이상인 경우에 집합 S의 비교되어진 path는 disjoint products를 생성하고 집합 P에 포함된다. 이것은 다음으로 비교되는 path가 집합 P의 모든 path와 서로 겹치지 않아야 하고, 이전 단계에서 비교된 paths와도 겹치지 않게 만들어야 하기 때문이다.

3. Numerical Example



<그림 1> 7-링크 네트워크

7개의 링크로 구성된 <그림 1>의 네트워크에서 링크 i 가 작동할 확률을 p_i , $i=1,\cdots,7$라고 할 때 JRI(i,j)를 계산하는 과정은 아래와 같다. 이 네트워크에 대하여 mps$=\{(26),(136),(147),(257),(1357),(1456),(2347)\}$이며, JRI(1,3)를 계산하는 경우, A_1, A_3, A_{13}, A_x을 minimal paths로부터 분류해보면 다음과 같다.

$$A_1 = \{(47),(456)\}, \qquad A_3 = \{(247)\},$$

$$A_{13} = \{(6),(57)\}, \qquad A_x = \{(26),(257)\}$$

A_x와 A_3을 비교하여 얻어지는 disjoint products는 $2\,4\,\overline{5}\,\overline{6}\,7$이며, $A_1 \cup A_x$와 A_3에서는 path (247)이 (47)에 포함(absorbing)되기 때문에 더 이상 비교할 수 있는 path가 A_3에 없으므로 얻을 수 있는 disjoint products는 없다. $A_1 \cup A_3 \cup A_x$와 A_{13}의 경우는 $\overline{2}\,\overline{4}\,6+\overline{2}\,4\,\overline{5}\,6\,\overline{7}+\overline{2}\,\overline{4}\,5\,\overline{6}\,7$이다. 결국 대응되는 reliability polynomial은 다음과 같은 식으로 표시된다.

$$\begin{aligned}\mathrm{JRI}(1,3) &= \overline{2}\,\overline{4}\,6+\overline{2}\,\overline{4}\,5\,\overline{6}\,7+\overline{2}\,4\,\overline{5}\,6\,\overline{7}-2\,4\,\overline{5}\,\overline{6}\,7 \\ &= (1-p_2)(1-p_4)p_6+(1-p_2)(1-p_4)p_5(1-p_6)p_7 \\ &\quad +(1-p_2)p_4(1-p_5)p_6(1-p_7)-p_2p_4(1-p_5)(1-p_6)p_7\end{aligned}$$

만일 $p_1=0.9$, $p_2=0.9$, $p_3=0.85$, $p_4=0.85$, $p_5=0.85$, $p_6=0.9$, $p_7=0.9$라고 하면 링크 1을 기준으로 나머지 다른 링크들과의 JRI(1, j), $j=2,\cdots,7$, 은 다음과 같다.

JRI(1,2) = −0.962561	JRI(1,3) = 0.0054675
JRI(1,4) = 0.0176175	JRI(1,5) = −0.0080325
JRI(1,6) = −0.00418625	JRI(1,7) = 0.0130262

참 고 문 헌

J.A. Abraham(1979), "An improved algorithm for network reliability", *IEEE Trans. Reliability, vol R-28, 58-61.*

M. J. Armstrong(1997), "Reliability-importance and dual failure-mode components", *IEEE Trans. Reliability, vol 46, 212-221.*

R. E. Barlow, F. Proschan(1981). "Statistical theory of reliability and life testing", *Holt, Rinehart, Winston.*

J. S. Hong, Chang Hoon Lie(1993), "Joint reliability-importance of two edges in an undirected network", *IEEE Trans. Reliability, vol 42, 17-23.*

M. O. Lock(1987), "A minimizing algorithm for sum of disjoint products", *IEEE Trans. Reliability, vol R-36, 445-453.*

M. O. Lock(1992), "Note on disjoint products algorithm", *IEEE Trans Reliability, vol 41, 81-84.*

J. M. Wilson(1990), "Improved Minimizing algorithm for sum fo disjoint products", *IEEE Trans Reliability, vol 39, 42-45.*