

구성요소가 서로 종속인 네트워크시스템의 신뢰성모형과 계산알고리즘⁺

(Reliability Modeling and Computational Algorithm of Network Systems with Dependent Components)

홍 정 식 (J. S. Hong)*

이 창 훈 (C. H. Lie)**

ABSTRACT

General measure in the reliability is the k -terminal reliability, which is the probability that the specified vertices are connected by the working edges. To compute the k -terminal reliability components are usually assumed to be statistically independent.

In this study the modeling and analysis of the k -terminal reliability are investigated when dependency among components is considered. As the size of the network increases, the number of the joint probability parameter to represent the dependency among components is increasing exponentially. To avoid such a difficulty the structured-event-based-reliability model (SERM) is presented. This model uses the combination of the network topology (physical representation) and reliability block diagram (logical representation). This enables us to represent the dependency among components in a network form.

Computational algorithms for the k -terminal reliability in SERM are based on the factoring algorithm. Two features of the factoring algorithm are the reliability preserving reduction and the pivoting edge selection strategy. The pivoting edge selection strategy is modified by two different ways to tackle the replicated edges occurring in SERM. Two algorithms are presented according to each modified pivoting strategy and illustrated by numerical examples.

1. 서론

네트워크 신뢰성은 네트워크로 이루어진 시스템의 신뢰성 관련 성능을 연구하는 분야이다. 가장 일반적인 척도로는 k -터미널 신뢰성(k -terminal

reliability)인데 이는 특정한 k 개의 vertex들이 작동 edge에 의해 서로 연결이 이루어질 확률을 의미한다. k -터미널 신뢰성 계산 문제는 "NP-hard"임이 밝혀져[3, 11], 최근의 연구는 특정 그래프에

* 서울산업대학교 산업공학과

** 서울대학교 산업공학과

+ 이 논문은 1988년도 한국과학재단 기초연구비에 의하여 연구되었음.

속하는 네트워크에 대한 polynomial 알고리즘을 개발하거나 그 부류의 네트워크조차 k-터미널 신뢰성 계산 문제가 “NP-hard”임을 입증하는 작업에 집중되고 있다. 그런데 이러한 연구 논문들은 모두 구성 요소간 독립(s-independence)을 가정하고 있다. 실제의 네트워크 시스템에서는 구성 요소들이 서로 위상적인 관계를 맺고 있음에 따라 근접한 구성 요소끼리 독립이 성립할 가능성은 매우 희박하다고 할 수 있다. 그런데 상호 관련된 고장(correlated failure)을 다룬 논문은 시스템이 2-unit(직렬 혹은 병렬)인 경우에 주로 되어있고 규모가 상대적으로 크며 구성요소간 상호위상적인 관계를 맺고 있는 특성을 지닌 네트워크에 대해서는 극히 적은 수의 연구가 수행되었다[5, 7].

그런데 네트워크 시스템에 있어 구성 요소간 종속을 고려한 상태에서 k-터미널 신뢰성을 계산하는 데는 2가지 문제점이 따른다. 즉 구성 요소가 증대될 때 이들간의 종속 관계를 표현하는 결합 확률 모수는 지수적으로 증가하므로 모수 추정(parameter estimation)과 계산 알고리즘의 효율성이라는 문제가 등장한다[5]. 이러한 문제점을 해결하기 위해서는 k-터미널 신뢰성 계산 알고리즘의 개발에 앞서 적절한 모형화가 필요하다. 모형화의 기본 접근 방법에 따라 이제까지의 연구는 크게 2가지로 나눌 수 있다. 첫째는 조건부 확률 모형[5, 8, 12]이고, 둘째는 독립 사건 도입 모형[4, 6, 7]이다.

첫째 접근 방법에 따른 모형에 관한 단점은 Lam과 Li의 논문[7]에 나와있다. 그리고 둘째 접근 방법에 따른 모형의 단점은 [15]에 나와있다.

우리는 본 논문에서 두 번째 접근 방법에 따른 모형과 이를 토대로 한 k-터미널 신뢰성 알고리즘을 제시하고자 한다. 이 모형은 구조적 사건에 근거한 신뢰성 모형(Structural Event based Reliability Model, SERM)으로 Lam과 Li의 모형을 특수한 경우로 포함한다.

2절에서는 새로운 모형을 제시하고 3절에서는 2절에서 제시된 모형을 토대로 k-터미널 신뢰성을

구하기 위한 알고리즘(독립을 가정한 k-터미널 신뢰성 계산 알고리즘의 수정)이 제시되며 4절에서는 실제로 알고리즘의 아이디어를 설명하고 5절에서는 새로운 모형과 알고리즘의 의미가 기술된다.

용어 정의

V : vertex들의 집합

E : vertex들의 무순서쌍으로 이루어진 edge들의 집합

$G(V,E)$: V 와 E 로 이루어진 무방향 단순그래프(undirected simple graph)

E_r : 중복 edge(replicated edge)들로 이루어진 집합

여기서 중복 edge라 함은 그래프 G 에서 서로 다른 vertex쌍으로 이루어진 edge에 동일한 물리적 의미를 띤 이름이 붙은 edge를 말한다. 따라서 E_r 은 두 개의 데이터 그룹으로 정확히 표현된다. 하나는 그래프 상의 모든 edge를 나타내는 vertex쌍들과 다른 하나는 이들 각 vertex쌍에 붙여지는 번호(이름)이다. 그림으로 나타낼 때 혼동이 야기되지 않는 범위 내에서 주로 후자만으로 E_r 을 표현할 것이다.

$Gr(V,E_r)$: V 와 E_r 로 이루어진 그래프

X_i : edge i 에 대한 상태 변수. edge i 가 작동(작동중지)이면 1(0)

Y_i : 독립 사건 i 에 대한 상태 변수. 독립사건 i 가 발생(미발생)이면 0(1)

$f(X)$: 시스템의 구조 함수(structure function)

K : V 의 부분 집합

$R_k(S)$: 그래프 G 에서 K 에 속하는 특정 vertex들이 서로 연결이 이루어질 확률

$G * i$: 그래프 G 에서 edge i 를 제거하고 i 의 양인접 vertex들을 서로 결합하여 생성된 그래프

$G - i$: 그래프 G 에서 edge i 를 제거한 그래프

E : edge들의 상태를 나타내는 상태 벡터

2. 사건에 근거한 구조적 신뢰성 모형 (SERM)

모형을 설명하기 전에 우선 다음과 같은 시스템 표현 방법을 고찰해 보자. 시스템 표현 방법은 그 의도하는 바에 따라 아래와 같이 나누어진다.

- (1) 형상적 표현(Physical Representation)
- (2) 논리적 표현(Logical Representation)

(1)은 시스템을 이루는 구성 요소와 전체 시스템의 실제적인 형태를 나타낸 것으로 이를 위해 위상 네트워크(network topology)가 주로 사용된다. 이것의 수학적 대상은 그래프이며 edge와 vertex가 둘 다 구성요소가 되며, 중복 edge는 나타나지 않는다.

(2)는 시스템에 있어서 어떤 특정한 임부의 수행을 목표로 할 때, 이를 달성하기 위해 구성요소들이 각각 어떻게 결합되어야 하는가, 혹은, 이와 반대로 시스템의 어떤 임부 수행 불능 즉 고장에 대해서 구성 요소들과 외부 사건들이 논리적으로 어떤 방식으로 결합하느냐를 나타내는 것이다. 전자의 목적으로 신뢰성 블록도(Reliability Block Diagram)가 사용되며 후자의 목적으로 Fault Tree가 흔히 사용된다. 그런데 위와 같은 2가지 표현 방법을 동시에 사용함으로써 네트워크 시스템에 있어 구성 요소간 종속성을 표현하는 새로운 방법을 생각해 볼 수 있다.

이 모형에서의 아이디어는 서로 종속인 edge들을 이의 작동중지에 관여하는 독립의 사건으로 대체하되, 가능한 종속관계를 표현하면서, 네트워크 형태의 입력을 변함없이 유지하고자 하는 것이다. 이를 위해 아래의 성질 2-1을 보자.

성질 2-1 중복 edge를 갖는 그래프 Gr은 모든 시스템의 논리적 표현을, 2-터미널 신뢰성 시스템 성공 확률로 하여, 나타낼 수 있다.

(증명) 주어진 시스템의 성공을 위한 최소 성공 집합을 A_1, A_2, \dots, A_r 이라 하자. 그러면 그래

프 Gr은 r개의 병렬 경로로 이루어지고 이들 각 경로는 해당 A_i 의 구성 요소를 edge로 한 직렬 경로이다. 이 때 시스템 성공 확률은 그래프 Gr에서의 2-터미널 신뢰성과 일치한다.

이에 따라 각 edge의 고장에 관여하는 서로 독립인 사건을 고려해보자. 이 경우 edge간의 종속 관계가 성립하는 것은 하나의 사건이 몇개의 edge의 고장에 동시에 영향을 미치기 때문이다. 따라서 주어진 edge와 이의 고장에 관련된 사건이 주어졌을 때 이들간의 논리적 관계를 성질 2-1에 따라 edge의 양 vertex를 source와 terminal로 하여 나타낼 수 있을 것이다.

이제 edge들이 서로 종속인 네트워크에 대한 신뢰성 모형은 edge들의 고장에 관여하는 서로 독립인 사건들을 고려함으로써 시스템의 위상 네트워크의 구성 요소를 이들 사건을 구성 요소로 하는 네트워크로 변환시키는 것이다. 이 경우 네트워크의 구성 요소의 하나인 edge i 는 그에 관여하는 사건들에 대해 $X_i = f(Y)$ 의 관계로 표현된다. $X_i = f(Y)$ 에 있어 이 구조 함수(structure function) f 의 표현은 성질 2-1에 따라 부분 그래프로 표현되며 이것으로 edge를 대체하면 우리는 서로 독립인 사건을 edge로 갖는 새로운 그래프 Gr을 얻는다.

이 과정을 문제의 입력 변환으로 보면 다음과 같다.

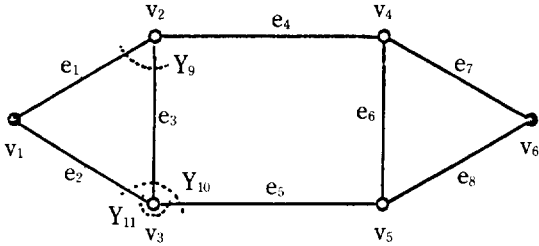
$$N = [G, P(X_1 = x_1, \dots, X_n = x_n), K]$$

$$X_i = f(Y)$$

$$N = [Gr, P(Y_i = 1), i = 1 \text{ to } n, K]$$

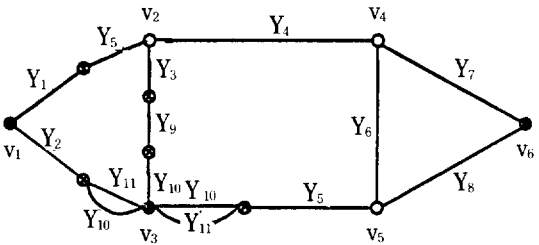
이 과정을 4절에서 다루어질 예제 네트워크를 통해서 보면 그림 2-1과 같다. 이 그림에서 까만 vertex는 K에 속하는 vertex이고 하얀 vertex는 K에 속하지 않는 vertex이며 \otimes 표시가 된 vertex는

실제적인 구성 요소로서의 vertex가 아니고 사건 간의 논리적 연결점을 표시해 주는 vertex이다.



$$\begin{aligned}
 G &= (V, E) & K &= (v_1, v_5, v_6) \\
 X_1 &= Y_1 \cdot Y_9 & X_2 &= Y_2 (Y_{10} \perp Y_{11}) \\
 X_3 &= Y_3 \cdot Y_9 \cdot Y_{10} & X_4 &= Y_4 \\
 X_5 &= Y_5 \cdot (Y_{10} \perp Y_{11}) & X_6 &= Y_6 \\
 X_7 &= Y_7 & X_8 &= Y_8
 \end{aligned}$$

(a) 종속인 구성 요소로 이루어진 네트워크와 관련 사건과 edge와의 관계



(b) 독립 사건을 구성 요소로 하는 $Gr(V, Er)$

<그림 2-1> 위상 네트워크와 신뢰성 불력도를 결합한 표현

3. SERM의 분석 및 계산 알고리즘

2절에서 제시된 모형에 따라 문제를 입출력 형식으로 표현하면 다음과 같다.

[PD] 입력 : $N=[Gr, P(Y_i=1), i=1 \sim n, K]$
 출력 : $R_k(G_r)$

구성요소값 독립을 가정한 네트워크에서 $R_k(G)$ 계산방법은 그 접근 방법에 따라 3가지 부류로 나눌 수 있다[1]. 그런데 이 중 가장 효율적이면서, 특수한 부류의 네트워크에 대해서는 polynomial 알고리즘을 제시해주는 분석 방법은 factoring 방법이라고 할 수 있다. 따라서 본 절에서는 factoring 방법을 활용하여 [PD]를 분석하여 $R_k(G)$ 를 계산해내는 알고리즘을 제시하고자 한다.

factoring 방법은 다음과 같은 식에 근거하고 있다.

$$\begin{aligned}
 R_k(G) &= P_i R_k(G/X_i=1) + (1-P_i) R_k(G/X_i=0) \\
 &= P_i R_k(G * 1) + (1-P_i) R_k(G - i)
 \end{aligned}$$

그런데 위의 수식이 중복 edge(replicated edge)를 갖는 Gr 에 있어서는 성립하지 않는다. 즉 여러개의 edge들에 논리적으로 하나의 사건이 해당될 경우 우리가 그 사건에 조건을 가할때, 그에 해당되는 모든 edge가 영향을 받기 때문이다. 가령 Gr 에 있어서 edge i 와 j 가 Y_k 를 나타낸다고 하자. 확률변수는 Y_k 이므로,

$$\begin{aligned}
 R_k(G) &= Pr(Y_k=1) R_k(G/Y_k=1) + Pr(Y_k=0) R_k(G/Y_k=0) \\
 R_k(G) &= Pr(Y_k=1) R_k(G * \{i, j\}) + Pr(Y_k=0) R_k(G - \{i, j\})
 \end{aligned}$$

여기서 $Y_k=1$ 이 관련 edge의 작동을 의미하도록 지시변수 Y_k 가 정해져 있다. 따라서 이러한 factoring 수식을 내적으로 이용한 신뢰성 보존 축소(reliability-preserving reduction) 또한 성립하지 않게 된다. 따라서 이러한 문제점을 다음 2가지 방법으로 해결해 볼 수 있다.

- (1) 모든 중복 edge에 조건을 가하는 다중-피벗
- (2) 기존의 factoring 알고리즘의 피벗 규칙(pivoting rule)의 변경

3-1. 다중 피벗에 따른 계산 알고리즘

Gr 에 중복 edge가 총 x 개 있다고 할 때, 다중

피봇을 적용할 경우 최악의 경우 2^x 개의 서로 다른 그래프 G^1, \dots, G^x 가 생겨날 것이다. 이들 각 그래프는 서로 독립인 edge들을 중복없이 구성요소로 갖고 있으므로 기존의 factoring 알고리즘이 곧바로 사용될 수 있다.

위의 아이디어에 따라 $R_k(Gr)$ 을 계산하는 절차를 단계별로 기술하면 다음과 같다.

단계 1. 검색 작업을 거쳐 그래프 Gr 의 중복 edge를 가려낸다.

단계 2. 중복 edge가 x 개 있을 경우 x 개의 중복 edge를 편의상 e_1, \dots, e_x 라 하자. 다중 피봇의 총 경우는 $E^1 = (e_1, \dots, e_x)$ 에서 $E^x = (\bar{e}_1, \dots, \bar{e}_x)$ 까지 2^x 개이다. 이에 따른 관련 edge들의 제거(deletion)와 융합(contraction) 작용을 하여 원 그래프 Gr 로부터 G^1, \dots, G^x 의 그래프를 발생시킨다.

단계 3. 단계 2에서 생성된 각 그래프를 대상으로 가장 효율적으로 알려진 factoring 알고리즘을 적용시켜 $R_k(G^i)$ 를 구한다.

단계 4. 단계 3에서 얻어진 $R_k(G^1), \dots, R_k(G^x)$ 로부터

$$R_k(Gr) = \sum_{i=1}^{2^x} \Pr(F_i) R_k(G^i) \text{의 수식으로부터}$$

$R_k(Gr)$ 을 구한다.

이 알고리즘에서 중복 edge의 수에 따라 단계 3의 Factoring 알고리즘 시행 횟수가 지수적으로 증가하므로 이 알고리즘의 효율성은 중복 edge의 수에 달려 있다고 할 수 있다. 즉 중복 edge의 중복도가 클수록 그리고 전체 중복 edge의 수는 적을수록 알고리즘이 효율적으로 문제를 풀어낸다. 이러한 방법은 edge의 중복도가 많은 fault tree(이 경우 edge는 basic event가 됨)의 분석 알고리즘의 개발에도 활용되었다[2]. 그런데 이러한 다중-피봇 방법을 쓰는 알고리즘은 기존의 factoring 알고리즘을 수정없이 서브루틴으로 사용하는 장점을

지니고 있으나 네트워크 크기가 증대될 경우 다중 피봇에 따른 edge들의 제거와 융합 작용에도 많은 시간이 소모되므로(중복 edge의 수가 많을 경우) 다중 피봇이 아닌 기존 factoring 알고리즘의 피봇 규칙을 변경함으로써 $R_k(Gr)$ 을 푸는 알고리즘이 더 효율적이 될 것이다.

3-2. 피봇 규칙 변경에 의한 알고리즘

factoring 알고리즘의 요체는 (1) 피봇 edge 선택 (2) 신뢰성 보존 축소이다. Factoring 알고리즘은 $R_k(G)$ 를 계산하고자 할 때 (2)를 먼저 수행한 후 더 이상 수행 불가능할 때 (1)을 시행하여 형성된 두 그래프 $G-i, G*i$ 에 대해서 다시 (2)를 수행하는 작업을 $R_k(G)$ 가 계산될 때까지 반복 수행한다. 그런데 중복 edge가 있을 경우 전술한 바와 같이 (2)를 쓸 수 없으므로 우선 비중복 edge에 대해 (2) 작업을 수행한 후에 피봇 규칙을 다음과 같이 변경한다.

규칙 1 : 중복 edge가 남아 있지 않은 경우 기존의 피봇 규칙을 사용한다. 만약 중복 edge가 있을 경우 중복도가 가장 큰 edge를 피봇 edge로 먼저 선택한다.

이 규칙을 적용하는 이유는 Gr 에서 축소를 수행하기 위해 중복 edge가 제거되어야 하는 면과 중복 edge는 이에 조건이 가해질 경우 관련 edge의 수가 많을수록 부분 그래프가 더욱 단순화된 그래프로 되는 측면을 이용하기 위해서이다. 이에 따른 알고리즘을 기존의 R.K.Wood 알고리즘을[13, 14] 기초로 단계별로 기술하면 다음과 같다.

단계 1. 신뢰성 보존 축소 수식에 따라 비중복 edge를 대상으로 신뢰성 보존 축소를 행한다. 그래프 Gr 이 단일 edge로 축소되면 멈춘다. 그렇지 않으면 단계 2로 간다.

단계 2. 수정된 피봇 규칙에 따라 edge i 를 선택한다. edge i 가 비중복 edge이면 단계 3으로

중복 edge이면 단계 4로 간다.

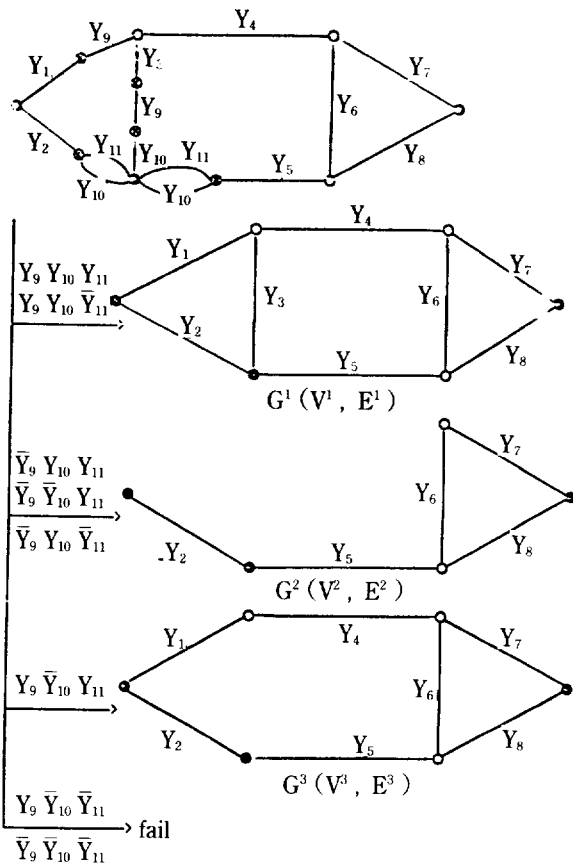
edge i 의 제거와 융합 작용을 거쳐 $G-i$ 와 $G*i$ 를 구축하고 단계 1로 되돌아 간다.

단계 3. edge i 의 제거와 융합 작용을 거쳐 $G-i$ 와 $G*i$ 를 구축하고 단계 1로 되돌아 간다.

단계 4. 중복 edge i 에 관련된 edge들의 집합을 C_i 라 하자. 이들 edge들의 제거와 융합 작용을 거쳐 $G-C_i$ 와 $G*C_i$ 를 구축하고 단계 1로 되돌아 간다.

4. 사례

그림 2-1의 예제 네트워크를 대상으로 3-1절과 3-2절의 알고리즘의 작동을 살펴보자. 우선 3-1절

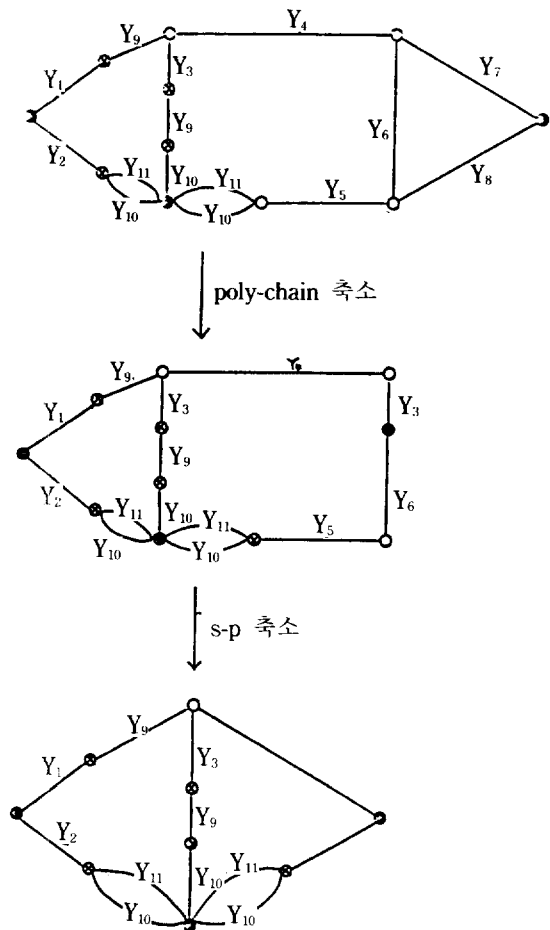


<그림 4-2> 주어진 그래프 G 에 다중 피벗을 적용할 경우 생성되는 그래프

의 알고리즘의 작동을 보자. 단계 1을 거치면 Y_9, Y_{10}, Y_{11} 의 사건 지시변수가 중복 edge에 해당됨을 알 수 있다. 따라서 단계 2를 적용하면 그림 4-1과 같은 부분 네트워크가 생성된다. 이들 네트워크에 단계 3과 단계 4를 거치면 $R_k(G)$ 가 구해진다.

$$R_k(G) = (p_9 p_{10} p_{11} + p_9 p_{10} q_{11})R_k(G^1) + (q_9 q_{10} p_{11} + q_9 q_{10} p_{11} + q_9 p_{10} q_{11})R_k(G^2) + (p_9 q_{10} p_{11})R_k(G^3)$$

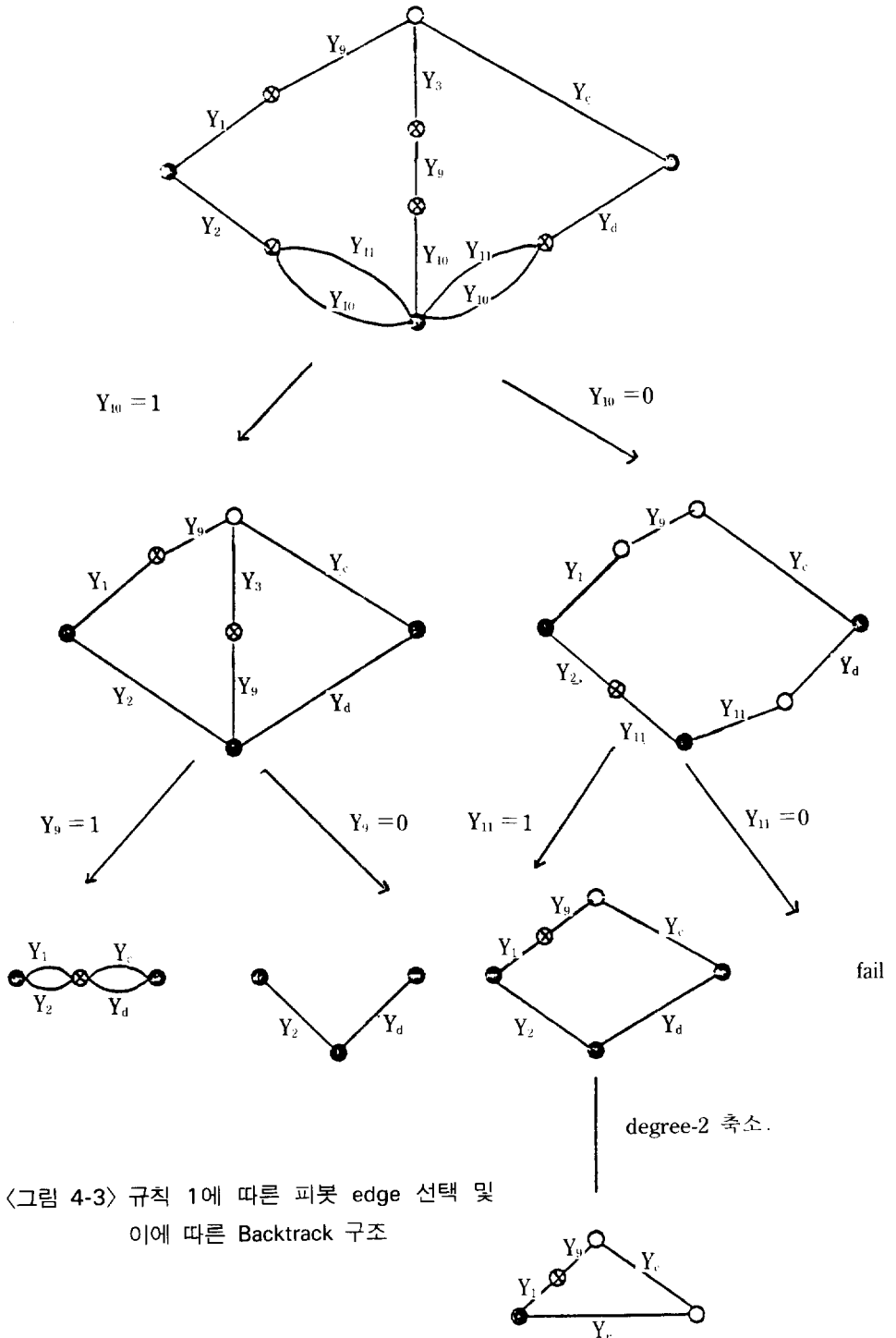
여기서 $R_k(G^1)$ 과 $R_k(G^2)$ 그리고 $R_k(G^3)$ 는 R.K. Wood[3]의 알고리즘을 쓰면 간단히 구해진다.



<그림 4-2> 비중복 edge를 대상으로 한 신뢰성 보존 축소

이제 3-2절의 알고리즘의 작동을 보자. 그림 2-1의 예제 네트워크에 단계 1을 적용하면 그림 4-2와 같은 네트워크가 나온다. 이제 더 이상 신뢰성

보존 축소가 불가능하므로 단계 2로 간다. 여기서 피벗 규칙에 따라 선택되는 edge는 Y_{10} 이라는 사건 지시변수에 해당되는 중복 edge이다.



<그림 4-3> 규칙 1에 따른 피벗 edge 선택 및 이에 따른 Backtrack 구조

따라서 단계 4로 가서 피벗을 한 후 단계 1로 가서 전 과정을 되풀이 한다. 이 과정에 따른 Backtrack 구조가 그림 4-3에 나와있다. 이에 따라 $R_k(G)$ 를, 기호 표현으로 쓰면 다음과 같다(실제 출력은 수치결과이나 계산 과정을 보이기 위해).

$$R_k(G) = \Omega_1 [p_{10} p_8 (p_1 \perp p_2) \cdot (p_c \perp p_d) + p_{10} q_9 p_2 p_d + q_{10} p_{11} \Omega_2 \left\{ \frac{p_2 p_d}{1 - q_2 q_0} \perp (p_1 p_9 p_c) \right\}]$$

여기서,

$$\Omega_2 = 1 - q_2 q_0$$

$$\Omega_1 = \frac{(\alpha + \sigma)(\beta + \sigma)}{\sigma}$$

$$p_c = \frac{\sigma}{\alpha + \sigma}$$

$$p_d = \frac{\sigma}{\beta + \sigma}$$

$$p_e = p_1 \cdot p_d$$

$$p_d = p_8 \cdot p_b$$

$$\alpha = q_9 p_8 p_6$$

$$\beta = p_9 q_8 p_6$$

$$\sigma = p_6 p_8 p_9 \left(1 + \frac{q_6}{p_6} + \frac{q_8}{p_8} + \frac{q_9}{p_9} \right)$$

5. 결론

본 논문에서는 구성 요소가 서로 종속인 네트워크에 있어 k-터미널 신뢰성 계산 문제를 다루는데 있어 적절한 모형화와 이에 따른 계산 알고리즘을 2가지 서로 다른 방식으로 제시하였다. 구성 요소 간 종속성을 고려하는데 있어 우리는 이들 구성 요소에 영향을 미치는 서로 독립인 사건을 도입하는 접근 방법을 따랐고 이 경우 가장 일반성을 띤 새로운 모형을 제시하였다. 이것은 신뢰성 불확도를 네트워크 시스템의 부분 네트워크로 변형 도입하는 아이디어에 근거하고 있다. 따라서 분석에서 다루는 네트워크는 중복 edge를 갖고 있는 그래프인 바, 3절에서 제시된 알고리즘은 본 논문에서 다루는 대상만이 아니고 일반 신뢰성 불확도를 대상으로 적용될 수 있다. 물론 적용되는 알고리즘의 효율성은 대상 네트워크에 있어 중복 edge의 중복도가 클수록 그리고 전체 중복 edge의 수가 적을수록 높아진다.

본 논문에서 제시된 모형화는 시스템 구성 요소들이 서로 인접한 요소들 사이에 종속이 부분적으로 나타나는 시스템(컴퓨터 네트워크, 통신 네트워크)에 활용될 수 있다.

참고문헌

1. Agrawal, A. and R. E. Barlow(1982), "A Survey of Network Reliability," ORC 83-5 Operations Research Center, Univ. of California, Berkeley.
2. Arnborg, S. (1981), "A Reduced State Enumeration-Another Algorithm for Reliability Evaluation", IEEE Trans. on Reliability, Vol.R-27, pp. 101-105.
3. Ball, M. O. (1986), "Computational Complexity of Network Reliability Analysis : An Overview", IEEE Trans. on Reliability, Vol. R-35, pp. 230-239.
4. Chae, K. C. and G. M. Clark (1986), "System Reliability in the Presense of Common-Cause Failure", IEEE Trans. on Reliability, Vol. R-35, pp. 32-35.
5. Hagstrom, J. N. and K. T. Mak (1985), "The Reliability Function and Dependent Component State Distribution", Univ. of Illinois, Chicago, working paper.
6. Heffes, H. and A. Kumar (1986), "Incorporating Dependent Node Damage in Deterministic Connectivity Analysis and Synthesis of Networks", Networks, Vol. 16, pp. 51-65.
7. Lan, Y. F. and V. O. K. Li (1986), "Reliability Modeling and Analysis of Communication Networks with Dependent Failures", IEEE Trans. on Communication COM-34, pp. 82-84.
8. Lin, P. M., Leon, B. J. and T. C. Hwang (1976) "A New Algorithm for Symbolic System Reliability Analysis", IEEE Trans. on Reliability, Vol.R-25, pp. 2-15.
9. Politof, T. and A. Satyanarayana (1986), "Efficient Algorithms for Reliability Analysis of Planar Networks-A Survey", IEEE Trans. on Reliability, Vol. R-35, pp. 252-259.
10. Politof, T. and A. Satyanarayana (1986), "Network Reliability and Inner-Four-Cycle-Free Graphs", MATH. of O. R., pp. 484-505.
11. Provan, J. S. (1986), "The Complexity of Reliability Computation in Planar and Acyclic Graphs", SIAM. J. Comput. pp. 684-702.
12. Spragins, J. and J. Assiri (1980), "Communication Networks Reliability Calculations with Dependent Failures", in Proc. Nat. Telecommun. Conf., pp. 25.2.1.-25.2.5.
13. Wood, R. K. (1982), "Polygon-to-Chain Reductions and Extentions for Reliability Evaluation on Undirected Networks", Ph. D. Dissertation, Univ. of California Berkeley, also Operations Research Center Report, ORC. 82-12.
14. Wood, R. K. (1986), "Factoring Algorithms for Computing K-terminal Network Reliability", IEEE Trans. on Reliability, Vol. R-35, pp. 269-278.
15. 홍정식 (1988), "중속인 구성요소들로 이루어진 네트워크의 신뢰성 모형 및 분석", 박사학위논문, 서울대학교.