內容을 고려한 無方向 네트워크의 信賴度計算
—Reliability Evaluation of a Capacitated Two-Terminal Network—

崔 明 鎬*
尹 德 均**

Abstract

This paper presents an algorithm CAPFACT to evaluate the reliability of a capacitated two-terminal network such as a communication network, a power distribution network, and a pipeline network. The network is good(working) if and only if it is possible to transmit successfully the required system capacity from one specified terminal to the other. This paper defines new capacitated series-parallel reduction to be applied to a Series-parallel structure of the network. New Capacitated factoring method is applied to a non-series-parallel structure. The method is based on the factoring theorem given by Agrawal and Barlow.

According to the existing studies on the reliability evaluation of the network that the capacity is not considered, the factoring method using reduction is efficient. The CAPFACT is more efficient than Aggarwal algorithm which enumerated and combined the paths. The efficiency is proved by the result of testing the number of operations and cpu time on FORTRAN compiler of VAX−11/780 at Hanyang University.

1. 序 論

1.1 硏究의 目的

Network의 信賴度(reliability) 計算은 통신 네트워크, 송전 네트워크, 송수(송유)관 네트워크의 設計 및 운용에 중요하다. 기존의 대부분의 硏究는 信賴度 尺度로 arc(幹線)의 연결성만 고려하였다. 信賴度 尺度는 arc의 方向의 有無와 규정된 node(頂點)의 個數에 따라 6가지로 나눈다.

① source-to-terminal-reliability
② source-to-k-terminal reliability
③ source-to-all-terminal reliability
④ two-terminal reliability
⑤ k-terminal reliability
⑥ overall reliability

현실적인 시스템에 있어서는 arc의 파라미터(parameter)로서 연결성 뿐만 아니라, 容量(capacity)도 고려해 줄 필요가 있다. 本 硏究의 模型은 容量을 고려한 two-terminal network의 信賴度 計算에 관한 것이다. 信賴度 尺度는 규정된 두 node가 단위 시간당 시스템의 주어진 흐름량을 arc의 容量 制限을 만족하면서 성공적으로 서로 주고 받을 確率이다. 硏究의 目的은 위의 模型에 관한 알고리즘 CAPFACT를 제시하는데 있다. 기존의 硏究는 arc의 파라미터(parameter)로 arc의 信賴度만 고려하였고, 容量은 고려하지 않았다.

1.2 先行 硏究 動向

1980년에 Lee[7]가 처음으로 容量을 고려한 네트워크의 信賴度 計算에 관한 模型과 알고리즘(algorithm)

*蔚山專門大學 工業經營科 敎授
**漢陽大學校 産業工學科 敎授
접수 : 1989. 6. 26.

을 제시하였다. 그는 사전식 정렬(lexicographic ordering)의 개념과 레이블링(labeling) 方法을 이용하였다. Lee[7]의 信賴度 尺度는 方向이 있는 Source-to-terminal reliability이 있다. 1982년에 Misra[8]은 Lee[7]의 알고리즘에 관하여 說明을 첨가하였다. 같은 해인 1982년에 Aggarwal[1, 2, 3, 4]은 Lee[7]의 模型이 方向이 있는 경우인데 반하여, 方向이 없는 경우 또는 方向이 있는 경우와 없는 경우의 혼합모형에도 適用할 수 있는, 보다 일반적인 알고리즘을 제시하였다. 本 硏究에서는 容量을 고려하고 方向이 없는 模型에 관하여 다룬다.

2. 理論的 考察

2.1 假定의 設定

(1) Network는 確率的으로 獨立인 arc들로 構成된다. 즉, 한 arc의 故障은 다른 arc의 故障에 영향을 미치지 않는다.

(2) 모든 arc의 方向은 없다.

(3) 시스템과 모든 arc은 作動 또는 故障의 두 狀態만 갖는다.

(4) 단위시간당 arc를 통한 흐름량은 arc의 容量에 의해 제한 받는다. 作動인 arc를 통해서는 단위 시간 동안 容量 만큼 흘릴 수 있고, 故障인 arc를 통해서는 조금도 흘릴 수 없다.

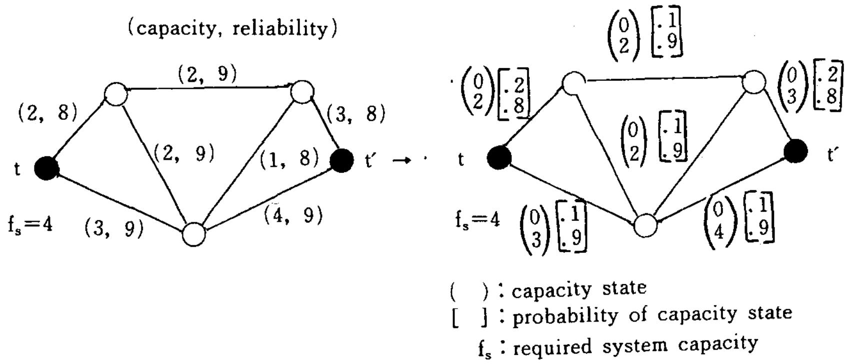
(5) 각각의 node에서 흐름량은 보존된다.

(6) 각각의 node의 容量에서 제한이 없다.

(7) Network는 Self-loop를 갖지 않는다.

2.2 定義의 設定

〈Fig. 1〉은 Aggarwal[3] 등이 제시한 例題에서 本 硏究의 模型에 적합하도록 方向이 있는 arc만 方向이 없는 arc로 고치고, 이를 容量狀態(capacity state)를 갖는 시스템으로 초기화한 것이다. 定義 1과 2를 設定하기 위해, X와 Y를 直列 또는 竝列構造를 이루는 arc의 容量狀態를 나타내는 確率變數(random variable)라 하고, 確率密度函數(pdf)는 容量狀態의 確率이라고 할 수 있다. 이의 確率密度函數를 f(X), f(Y)로 나타내자. 그러면, 集合 X, Y를 아래와 같이 나타낼 수 있다.



Figuire 1. Initialization of a Given Network

$$X=\{x_1,\ x_2,\cdots\cdots,\ x_i,\cdots\cdots,\ x_e\},\ 0<f(x_i)\leqslant 1,\ \sum_{i=1}^{l} f(x_i)=1$$

$$Y=\{y_1,\ y_2,\cdots\cdots,\ y_j,\cdots\cdots,\ y_m\},\ 0<g(y_j)\leqslant 1,\ \sum_{j=1} g(y_j)=1$$

그리고, 集合 S_k, S_k'는 다음과 같다.

$S_k=\{(i,j)\mid Z_k=\min\{\min(X_i,\ Y_j),\ f_s\}$를 만족하는 모든 $(i,j)\}$

$S_k'=\{(i,j)\mid Z_k=\min(X_i+Y_j,\ f_s)$를 만족하는 모든 $(i,j)\}$

단, f_s는 arc의 흐름량

Z는 容量 狀態

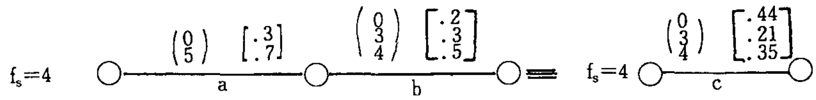
(i, j)는 node i로부터 node j의 arc

(1) 定義 1 : 容量을 고려한 直列縮小(Capacitated Series Reduction)

arc의 直列構造란 次數(degee)가 2인 node에 부속한 arc들의 構造를 말하고, ⊗를 容量을 고려한 直列縮小 演算字로 나타낼 때, 定義1에 의하여 다음의 容量狀態 Z와 그의 確率 $h(Z_k)$를 갖는 하나의 arc로 縮小된다. 식은 다음과 같다.

$$Z = X \otimes Y = \{Z_1,\ Z_2, \cdots\cdots,\ Z_i, \cdots\cdots, Z_n\}$$
$$= [\min\{\min(x_i,\ y_j),\ f_s\},\ \text{for all}(i,j)]$$
$$h(Z_k) = \sum_{(i,j)\in s} f(x_i) \times g(y_j),\quad 0 \leqslant h(z_k) \leqslant 1,\quad \sum_{k=1} h(z_k) = 1$$

〈Fig. 2〉는 直列構造를 이루는 arc a와 b를 하나의 arc c로 縮小시킨 例를 나타낸다.



a ⊗ b		b		
		[.2] (0)	[.3] (3)	[.5] (4)
a	[.3] (0)	(0) [.06]	(0) [.09]	(0) [.15]
	[.7] (5)	(0) [.14]	(3) [.21]	(4) [.35]

→ c $\begin{pmatrix}0\\3\\4\end{pmatrix}$ $\begin{bmatrix}.44\\.21\\.35\end{bmatrix}$

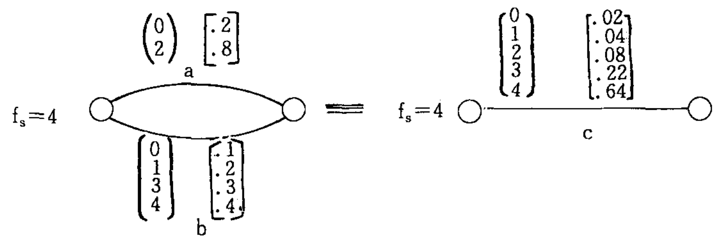
Figuire 2. Example of Capacitated Series Reduction

(2) 定義 2 : 容量을 고려한 竝列縮小(Capacitated Parallel Reduction)

arc의 竝列構造란 동일한 두 끝 頂點(end-node)을 갖는 arc들의 構造를 말하고, ⊕를 容量을 고려한 竝列縮小 演算字로 나타낼 때, 定義 2에 의하여 다음의 容量狀態 Z와 그의 確率 $h(z_k)$를 갖는 하나의 arc로 縮小된다.

$$Z = X \oplus Y = \{z_1,\ z_2, \cdots\cdots,\ z_i, \cdots\cdots,\ z_n\}$$
$$= [\min\{X_i + Y_j\},\ f_s\},\ \text{for all}(i,j)]$$
$$h(z_k) = \sum_{(i,j)\in s} f(x_i) \times g(y_j),\quad 0 \leqslant h(z_k) \leqslant 1,\quad \sum_{k=1} h(z_k) = 1$$

〈Fig. 3〉은 竝列構造를 이루는 arc a와 b를 하나의 arc c로 縮小시킨 例를 나타낸다.



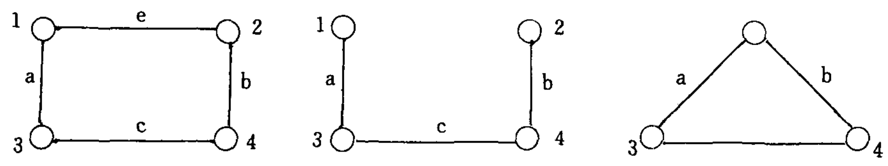
a ⊕ b		b			
		[.1] (0)	[.2] (1)	[.3] (3)	[.4] (4)
a	[.2] (0)	(0) [.02]	(1) [.04]	(3) [.06]	(4) [.08]
	[.8] (2)	(2) [.08]	(3) [.16]	(4) [.24]	(4) [.32]

→ $\begin{pmatrix}0\\1\\2\\3\\4\end{pmatrix}$ $\begin{bmatrix}.02\\.04\\.08\\.22\\.64\end{bmatrix}$

Figuire 3. Example of Capacitated Parallel Reduction

2.3 arc의 削除(deleting)와 短縮(contracting)

arc e의 削除는 e만 network G에서 없애고, 두 끝 頂點(end-node)을 남겨두는 것을 말한다. arc e의 短縮은 Network G에서 없애고, e의 두 끝 頂點을 합쳐 하나의 超頂點(supervertex)으로 만드는 것을 말한다. 〈Fig. 4〉에서 보여주고 있다. (b)는 (a)의 arc e를 削除시킨 것을 나타내고, (c)는 短縮시킨 것을 나타낸다.



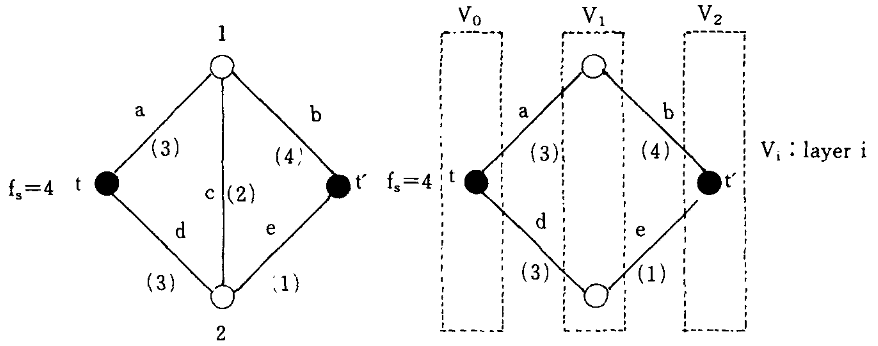
(a) A Given Network (b) Deleting of Edge e (c) Shorting of Edge e

Figuire 4. Deleting and Shorting of Edge e

2.4 층화 네트워크(layered network)

arc의 短縮이 존재하지 않는다면 규정된 두 node를 연결하는 경로를 통하여 흐름량을 흘림으로서 단위 시간당 시스템의 주어진 흐름량을 경로를 통한 흐름량 만큼 줄일 수 있다. Dicnic[6]은 최대 흐름량 문제(maximal flow problem)를 푸는 과정에서 층화네트워크를 사용하였다.

층화네트워크(layered network)의 층i를 V_i로 나타내자. 각각 $1 \leqslant i \leqslant l$에 대하여, 층 v_{i-1}의 node로부터 층 v_i의 node에 이르는 모든 arc들의 集合을 층화네트워크의 切斷集合(cuset)이라고 말하고 E_i로 나타낸다. 이 切斷集合을 이루는 arc들의 容量狀態의 合이 단위 시간당 시스템의 주어진 흐름량 보다 작다면 시스템은 故障의 狀態에 놓이게 되므로, 더이상 信賴度를 구하려고 分解할 필요없이 패덤(fathom)된다. 〈Fig. 5〉의 E_1의 容量狀態의 合은 6이고, E_2의 容量狀態의 合은 5이다.



(a) Bridge Network (b) Layered Network of (a)

Figuire 5. Example of a Layered Network

2.5 分解法(Pivotal Decomposition or Factoring Method)

(1) 容量을 고려하지 않은 分解法

R(G)를 Network G의 信賴度, P_e를 arc e가 作動할 確率이라고 하자. Network G의 상태는 한 arc e의 作動 또는 故障에 따라 다음의 두 狀態로 나누어 진다.

$$R(G)=P_e \cdot R(G \mid e\text{가 作動})+(1-P_e) \cdot R(G \mid e\text{가 故障}) \quad \text{①}$$

식 ①은 pivotal decomposition으로 유명하며, 일찌기 Moskowitz[9]에 의하여 信賴度 計算에 이용되었다. G_e를 G의 arc e를 短縮함으로써 구한 部分네트워크(shubnetwork)라고 하고, G_{-e}를 G의 arc e를 削除함으로써 구한 部分네트워크라 하자. 그러면, 식 ①은 다음과 같이 표현된다.

$$R(G)=P_e \cdot R(G_e)+(1-P_e) \cdot R(G_{-e}) \quad \text{②}$$

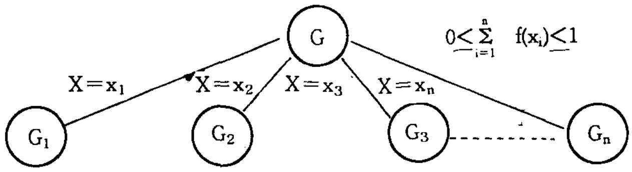
식 ②를 分解整理(factoring theorem)라고 하는데, Satyanarayana[10] 등에 의하면 容量을 고려하지 않은 無方向 네트워크의 信賴度 計算에 중요하고 효율적으로 이용된다.

(2) 容量을 고려한 分解法

容量을 고려한 경우에는 直列-竝列 縮小에 의하여 容量狀態가 2 以上일 수 있다. 식 ①과 ②는 고려한 경우에, X_{max}를 확정적이 아닌(容量狀態의 個數가 2 以上인) 어떤 arc의 容量狀態의 最大値라 할 때, 다음과 같이 새로운 식으로 표현된다.

$$R(G)=f(X=X_{max})\cdot R(R \mid X=X_{max})+(1-f(X=X_{max}))\cdot R(G \mid X=X_{max}) \quad \cdots\cdots ③$$

이때, $R(G \mid X=X_{max})$ 또는 $R(G \mid X=X_{max})$를 구할 때 arc의 短縮 또는 削除가 가능하면 이를 수행한다. 이에 따라, 直列-竝列 構造를 찾아 縮小 가능하면 縮小를 수행하여 고려해야 할 네트워크의 크기를 줄여 나간다. 식 ③을 계속 적용해 나가면, 그 構造는 결국 트리構造(tree structure)를 이루게 된다. 〈Fig. 6〉은 arc의 分解構造를 나타낸다. 이때 分枝되는 部分네트워크(subnetwork)들은 서로 배반이다. 트리(tree)는 네트워크의 특수한 경우이다. $|e|$를 네트워크의 arc의 數라고 할 때, 分解構造는 최악의 경우에 $2^{|e|}$의 잎(left)을 갖는 트리(tree)가 된다. 그러나, 연속적인 또는 교대적인 容量을 고려한 直列-竝列 縮小를 수행하고, 층화네트워크에 의하여 가장 적은 arc를 갖는 경로의 arc들에 대하여 분해하고 切斷集合의 容量狀態의 合을 구하여 패덤(fathom) 가능성을 검사함으로써 잎(leaf)의 數를 줄일 수 있다.



$G_1, G_2, G_3, \ldots, G_n$: Subnetworks of G

Figuire 6. Decomposition Structure of a Capacitated Edge

3. 알고리재 CAPFACT와 Aggarwal의 比較分析

3.1 CAPFACT의 適用例

〈Fig. 7〉은 〈Fig. 1〉의 초기화된 問題를 本 硏究의 CAPFACT에 의한 풀이과정을 나타낸다.

3.2 Aggarwal의 알고리즘

(1) Aggarwal의 硏究範圍

Aggarwal[3]등은 問題를 다음과 같이 3部分으로 나누고, ①과 ③은 기존연구를 참조하도록 하였고, ②의 과정만 제시하였다.

① 그래프의 연결행렬로 부터 최소경로(minimal path)를 모두 구한다.

② 최소경로들을 서로 조합하여 단위 시간당 시스템의 주어진 흐름량을 만족하는 타당그룹(valid group)들을 구한다.

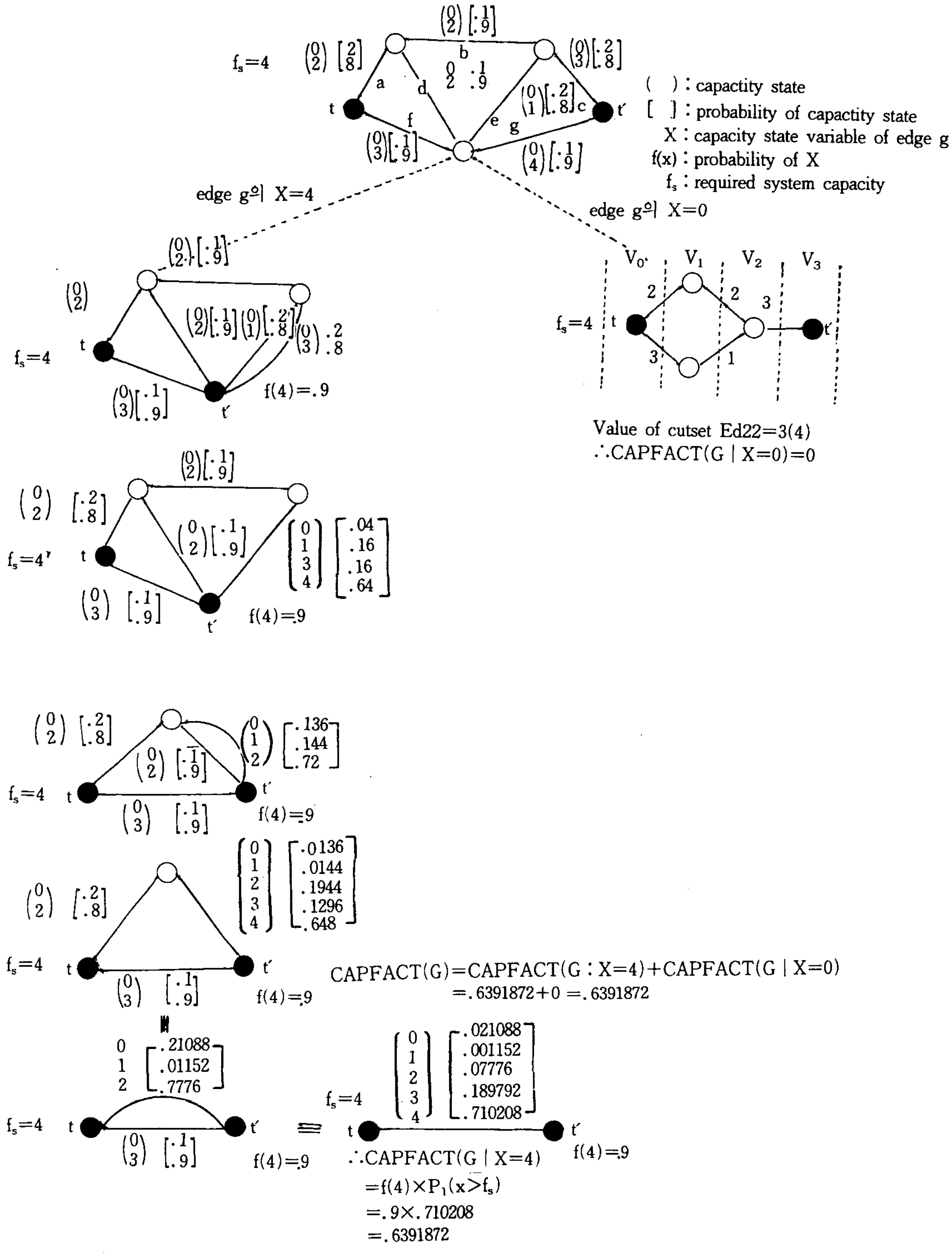
③ 부울대수에 의하여 ②의 타당 그룹들을 서로 배반이 되도록 하여 시스템의 信賴를 유도한다.

(2) Aggarwal algorithm의 問題點

無方向 네트워크인 경우 (1)의 ①에서 최소경로를 구할 때 방문하였던 arc를 다시 방문할 수 있고, 사이클(Cycle)이 발생할 수도 있기 때문에 최소경로를 모두 구하는 것은 어렵다. 그리고 (1)의 ③에 관한 기존의 알고리즘의 Complexity는 최악의 경우 non-polynomial Time이다.

(3) CAPFACT의 効率性

CAPFACT에 의하면, 방문의 중복이나 사이클을 고려해 줄 필요없다. 또한 問題의 크기를 줄일 수 없으면 分解法에 의하여 줄일 수 있도록 한다. 分解法에 의한 分解構造은 트리構造(tree structure)를 이루게 되는데 서로 배반이 사상으로 직접 분해되므로 부울대수에 의한 배반사상으로 나누는 과정이 필요없다.



Figuire 7. Example Solved by the Proposed CAPFACT

Aggarwal[3]은 (1)의 ②과정만 푸는 프로그램을 제시하였는데, ⟨Fig. 7⟩의 예를 푸는데 ①에 의해 7개의 경로가 이미 주어졌을 때, 이 7개의 경로를 182번 서로 조합하고 비교하여 타당그룹을 구하였다. 결국, 이들 위해서만 arc의 數가 7개 이므로 7×182=1274번의 比較 演算이 필요하다. CAPFACT에 의하면, 短縮된 部分네트워크의 과정에서는 縮小에 의하여 問題의 크기를 줄여나가 결국 t와 t′를 연결하는 하나의 arc로 간단히 표현된다. 削除된 部分네트워크에서는 층화네트워크의 容量狀態 合을 검사함으로써 패덤되어 더 이상 고려할 필요없다.

VAX-11/780의 사용하고, FORTRAN Compiler로 실행시켜 본 결과, 入出力의 과정을 뺀 CPU time이 CAPFACT에 의하면, 시스템의 信賴度를 구하는데 0.03초 소요되었고, Aggarwal에 의하면 ②의과정만 푸는데 0.23초 소요되었다.

4. 結　論

本 硏究에서는 보다 현실적인 模型인 容量을 고려한 two-terminal network의 信賴度 計算에 관한 알고리즘 CAPFACT를 제시한다. 容量을 고려한, 새로운 直列-竝列 縮小를 定義하여 縮小에 의한 分解法을 가능하게 한다. 分解回數를 줄이기 위해 필요에 따라 층화네트워크를 만든다.

Aggarwal[3]은 本 硏究의 模型에 적용가능한 알고리즘을 제시하였는데, 경로를 열거하고 조합하는 方法을 사용하였다. 간단한 例題를 컴퓨터로 실행시켜 본 結果, CAPFACT가 Aggarwal이 제시한 알고리즘 보다 계산소요시간상 效率的임이 입증된다. 그리고 기존의 容量을 고려하지 않은 模型은 本 硏究의 特殊한 경우로서, 단위 시간당 시스템의 주어진 흐름량이 1이고, 모든 arc의 容量이 1인 경우이다.

앞으로, 容量을 고려한 k-terminal reliability, overall reliability, 方向이 있는 네트워크의 信賴度 計算에 관한 硏究가 요구되며, 本 硏究에서 제시한 定義1과 定義2가 유효하게 적용되리라 기대된다.

References

1. Aggarwal, K. K., Gupta, J. S. and Misra, K, B.(1975), "A Simple Method for Reliability Evaluation of a Communication System", IIIE Trans. *Reliability*, vol. 23, 563-566, May.
2. Aggarwal, K. K., Gupta, J. S. and Misra, K. B.(1975), "Reliability Evaluation, a Comparative study of Different Techniquen", *Microelectronics & Reliability*, vol. 14, 49-56.
3. Aggarwal, K. K., Bajwa, J. S. and Chopra, Y. C.(1982), "Capacity Consideration in Reliability Analysis of Communication System", IIIE Trans. *Reliability*, vol. R-31, No. 2, 177-181, June.
4. Aggarwal, A. and Barlow, R. E.(1984), "A Survey of Network Reliability and Domination Theory", *Operation Research*, vol. 32, No. 3, 478-492.
5. Deo, Nar Sing(1974), *Graph Theory with Application to Engineering and Computer Science*, Prentice-Hall, Inc., Englewood cliffs, N. J.
6. Dinic, E. A., "Algorithm for Solution of a Problem of Maximal Flow in a Network with Pouser Estimation", *Soviet Math. Dok 1.*, vol. 11, 1277-1280.
7. Lee, S. H.(1980), "Reliability Evaluation of a Flow Network", *IIIE Trans. Reliability*, vol. R-29, No. 1, 24-26, April.
8. Misra, K. B. and Pramod Prasad(1982), "Correspondence", *IIIE Trans. Reliability*, vol. R-31, No. 2, 174-176, June.
9. Moskowitz, F., "The Analysis of Reduncancy Network", *AIEE Trans. Commun. Electron.*, vol. 39, 627-632.
10. Satyanarayana, A. and Chang, M. K.(1983), "Network Reliability and the Factoring Theorem", *Networks*. vol. 13. 107-120.