

單純經路搜索의 프로그래밍 및 그 應用에 關한 研究(綜合) —Studies on the Computer Programming for Searching the Simple Paths and Its Applications(Summary)—

鄭 秀 一 *

ABSTRACT

The former paper (Part I) studies two methods for searching and determining the simple paths in an acyclic or a cyclic network. The two methods are computer programmed as subroutines (SPLP1 and SPLP2) for various use. And a few examples of its applications are discussed.

Another paper (Par II) studies the reliability computation for a network by using the Event Space Methods. A computer program is developed for the computation by applying the SPLP2 subroutine subprogram.

In this paper, the former results are summarized with another computer program for reliability computation by using the Path Tracing Methods. The two subroutines appear in the Appendix as reference for others. The programs can be used in the reliability computation of reducible and irreducible structure networks.

I. 序 論

Operations Research 분야의 Network 문제 중 最長經路問題는 建築·土木工事, 大單位裝備의 維持, 新製品의 研究 開發 등 복잡한 作業活動에 對한 計劃, 日程의 作成 및 管理에 PERT 또는 CPM 등의 技法으로 활발히 應用되고 있다. 그러나 이와는 正反對의 개념을 가지는 最長經路問題는 裝備의 交替, 投資計劃, 旅程 또는 set-up 問題 등에 있어서의 所要經費 및 所要時間의 최소화 등 광범위한 應用可能性이 있음에도 불구하고 그

活用이 매우 지지부진한 상태이다. 그래서 필자는 앞으로의 應用開發에 대비하여 여러가지 형태의 最長經路決定을 위한 algorithm 들을 조사하여 각 algorithm의 計算能率을 比較·考察한 바가 있다.^{2),3),4)}

그런데 Network 問題에 있어서 좀더 單純하고 素朴한 개념인 單純經路 — 特定の 한 node에서 다른 한 node에 이르는 모든 可能한 經路 — 의 搜索에 대해서는 흔히 그 單純性 때문에 그리고 또한 그 번잡성 때문에 이에 대한 研究·考察이

*仁荷大學校 工科大學 産業工學科 副教授

거의 무시되어 온 실정이다. 그러나 computer의 급속한 보급과 處理能力的 增大로 인하여 變 質성 문제는 일단 解消되었다고 할 수 있을 것이다. 그래서 필자는 單純經路 搜索方法을 調査·改 善하여 2가지 單純經路搜索方法을 computer pro- gramming해서 最長短經路의 搜索에 應用⁵⁾ 해 보았으며, 또한 事象空間法에 의한 信賴度計算에 도 利用⁶⁾해 보았다.

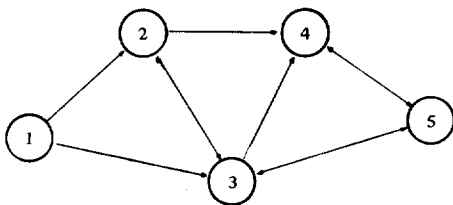
最近 micro computer의 보급이 급격히 증가 되면서 computer program도 FORTRAN 또는 COBOL 등의 言語보다는 BASIC 言語로 作 成되는 傾向이 나타나고 있는 바, 필자는 前述한 2가지 單純經路搜索方法을 BASIC 言語로 再 programming하여 이를 利用할 수 있는 여러 分 野에 應用·檢討해 보았다.

Cyclic 또는 acyclic network에 있어서의可 能한 모든 單純經路를 microcomputer를 利用하 여 찾아내고자 하는 이들의 便宜를 위해 前述한 2가지 單純經路搜索方法을 SPLP1 및 SPLP2 라는 명칭의 subroutine subprogram으로 作 成하여 附錄에 게재하였다. 그러나 事象空間法 및 經路追跡法에 의한 信賴度計算 program은 紙面 關係로 부득이 수목을 省略하였다. 필자가 使用한 micro computer는 Tandy Radio-Shack 80 M- icro Computer임을 參考로 밝혀둔다.

II. 單純經路搜索 및 그 應用의 프로그램

1. 單純經路搜索의 Subroutine Subprogram

그림 1의 network를 例로 들어 單純經路搜索 의 方法 및 computer program의 內容을 설명 해 보고자 한다. 出發點을 node 1, 終着點을 node n (여기에서는 node 5)이라고 하고 node 1에 서 node n 에 이르는 可能한 모든 單純經路를 搜索하고자 한다.



(그림 1) Cyclic network의 例

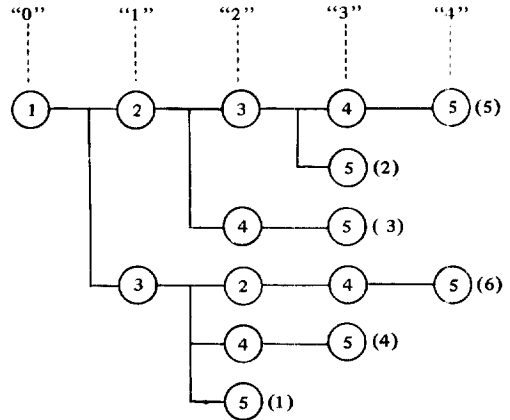
(1) Subroutine SPLP 1

우선 누구나 생각해 보게 되는 單純하고 直感的인 方法으로 network의 內容을 그림 2와 같은 from-to table로 作成한다. 그림 2에서 •은 node i 에서 node j 로 갈 수 있는 經路(arc)가 있음을, 빈칸은 그러한 經路가 없음을 나타내고 있다.

to from	1	2	3	4	5
1		•	•		
2			•	•	
3		•		•	•
4					•
5			•	•	

(그림 2) 그림 1의 network에 대한 from-to table

우선 出發點 node 1에서 꼭 1개의 arc를 통 해서 갈 수 있는 node는 node 2 및 3이다. 이 들은 그림 3에서 "1"로 표시되어 있는 node들



- (1) 1 - 3 - 5
- (2) 1 - 2 - 3 - 5
- (3) 1 - 2 - 4 - 5
- (4) 1 - 3 - 4 - 5
- (5) 1 - 2 - 3 - 4 - 5
- (6) 1 - 3 - 2 - 4 - 5

(그림 3) Subroutine SPLP1의 單純經路 搜索內容

이다. 다음 꼭 2개의 arc들을 통해서 到達할 수 있는 node들은 node 3 및 4, 그리고 node 2, 4 및 5로서 이들은 그림 3에 "1"로 표시되어 있는 node 2 및 3으로부터 또 하나의 arc를 통해서 갈 수 있는 node들로서 node 2로부터의 node 3 및 4, 그리고 node 3으로부터의 node 2, 4 및 5이다. 이들은 그림 3에 "1"로 표시되어 있는 node들로부터 그림 2를 조사하여 쉽게 찾아낼 수 있으며 그 결과가 그림 3에 "2"로 표시되어 있다.

그러면 1-3-5라는 하나의 單純經路를 찾아낼 수가 있다. 이것은 제일 먼저 搜索된 單純經路로서 그림 3에 (1)로 표시되어 있다.

한번 經由한 node들은 다시 經由하지 않는다는 條件下에서 꼭 3개의 arc들을 통해서 到達할 수 있는 node들은 node 3으로부터의 node 4 및 5, node 4로부터의 node 5, node 2로부터의 node 4, 그리고 node 4로부터의 node 5로서 이들은 그림 3에 "3"으로 표시되어 있으며 이때 1-2-3-5 (2), 1-2-4-5 (3) 및 1-3-4-5 (4)라는 두번째, 세번째 및 네번째 單純經路들이 搜索된다.

이러한 過程을 반복하면 주어진 network에서의 가능한 모든 單純經路들이 搜索될 수 있다. 그림 1의 network에 대한 node 1에서 node 5에 이르는 모든 單純經路들의 搜索結果가 그림 3의 아랫부분에 (1)~(6)으로 정리되어 있다. 이들을 檢討해 보면, 이 方法에 의해 單純經路를 搜索하는 경우 單純經路들이 通過하는 arc의 數가 적은 것부터, 그리고 같은 數인 경우에는 앞쪽의 node番號가 적은 것부터 搜索되게 된다는 特徵이 있다.

以上の 方法을 computer programming한 結果가 附錄에 실려 있는 Subroutine SPLP 1이며 이의 使用方法은 remarks部分을 參照하면 쉽게 理解될 것이므로 여기서는 그 內容을 省略한다.

이 方法을 利用하여 單純經路를 搜索하는 경우 그림 2에서의 from-to table 作成에서 상당한 dimension을 設定하게 되어 곤란하지만 data의 入力 및 變更이 容易하며 또한 應用時 關聯事項과의 連結이 쉬워진다는 長點이 있다.

終着點인 node n (여기에서는 node 5)에서 node j ($j \leq n$)에 이르는 arc는 實際에 있어서는 必要없는 data이므로 from to table 作成時에

省略하는 것이 바람직하다.

(2) Subroutine SPLP 2

前述한 搜索方法의 短點이라고도 할 수 있는 상당한 크기의 dimension 使用 및 from-to table에서 node i 로부터 node j 에 이르는 arc가 있는지의 여부를 가리는 데에 필요한 比較回數를 될 수 있는 한 줄이는 方法이다.

卓永奉⁷⁾은 D. Kroft⁹⁾의 搜索方法을 改善하여 그림 1의 network에 대해 1, 2, 3, 0, 2, 3, 4, 0, 3, 2, 4, 5, 0, 4, 5, 0, 5, 3, 4로 나타내게 되는 network definition data를 作成하여 利用하였다.

여기에서 0은 delimiter로서 network의 內容을 나타내는 data를 作成하여 利用하였다. 여기에서 0은 delimiter로서 network의 內容을 나타내는 data의 群을 區分하는 역할을 하며 各群內에서의 data는 첫째 숫자의 番號를 갖는 node로부터 delimiter인 0의 바로 앞까지의 나머지 숫자의 番號를 갖는 node(들)로 갈 수 있는 가능한 arc가 있음을 나타내고 있다.

그러나 여기에서는 이를 더욱 簡略化하여 搜索의 效率를 더욱 向上시켰다. 즉 特定の node i 에서 다른 node (s) j 로 連結되는 arc가 있을 경우에는 delimiter인 0를 除去하는 代身 node i 의 番號에 "-"符號를 붙여 중으로서 入力 data의 量을 줄이고, 그러므로서 單純經路의 搜索時에 比較回數를 줄여 搜索效率를 向上시켰다. 아마도 network의 內容을 이 이상 더 간단히 나타낼 수 있는 方法은 없을 것으로 생각된다.

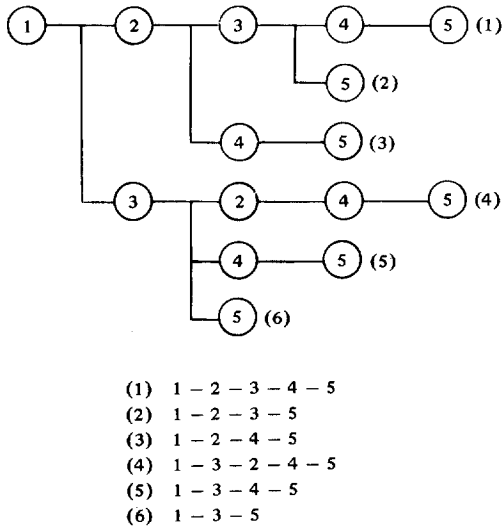
이와같은 入力 data를 利用하여 單純經路를 搜索하는 方法을 설명하면 다음과 같다.

우선 그림 1의 network에 대해서 그림 4와 같은 入力 data를 作成한다. 出發點인 node 1에서 꼭 1개의 arc를 통해서 갈 수 있는 node들은 node 2 및 3이나 node 2만을 고려하여 우선 1-2라는 未完成 單純經路를 찾을 수 있다. Node 2에서 1개의 arc를 이용해서 到達할 수 있는 node들은 node 3과 4이다. 이들 node 3 및 4는 그림 4의 入力 data를 맨처음부터 다시 調査하여 -2를 찾아내어 그에 잇달은 "-"符號가 붙어 있지 않은 node 番號들을 찾으면 될 것이다. 여기에서도 역시 node 3만을 고려하여 1-2-3이라는 未完成 單純經路를 作成해 놓는다.

-1, 2, 3, -2, 3, 4, -3, 2, 4, 5, -4, 5, -5, 3, 4

(그림 4) 그림 1의 network에 대한 簡略化된 入力 data

한번 經由한 node 들은 다시 經由하지 않는다는 條件下에서 이와같은 過程을 반복하면 제일 먼저 1-2-3-4-5라는 하나의 完全한 單純 經路를 搜索해 낼 수가 있다. 제일 먼저 搜索된 單純 經路이므로 앞에서와 같이 그림 5에 (1)로 나타내었다.



(그림 5) Subroutine SPLP2의 單純 經路 搜索内容

다음에는 위에서 搜索된 單純 經路를 利用함으로써 出發點인 node 1로부터 搜索을 시작하는 번거로움을 줄일 수 있다. 즉 單純 經路 (1)에서의 node 4에서는 더 이상 連結될 수 있는 node가 그림 4를 檢討해 보면 없으므로 單純 經路 (1)에서 node 4의 바로 앞 node인 node 3에서 連結될 수 있는 node를 찾으면 node 5가 될 것이고 이로서 또 하나의 새로운 單純 經路인 1-2-3-5(2)를 搜索해 낼 수가 있다. 다시 이 單純 經路 (2)를 利用하면 앞에서와 같은 節次에 의해 1-2-4-5(3)라는 세번째의 單純 經路를 찾아 낼 수가 있다.

以上과 같은 過程을 반복하면 주어진 network에서의 可能한 모든 單純 經路들이 搜索될 수 있다. 그림 1의 network에 대해서 node 1에서 node 5에 이르는 可能한 모든 單純 經路들을 搜索한 結果가 그림 5의 아랫부분에 (1)~(6)으로 정리되어 있고 그것은 搜索된 順序만 다를 뿐 그림 3에서의 結果와 마찬가지이다. 이 搜索方法에 의해 單純 經路가 搜索되는 順序는 아라비아 숫자의 順序를 alphabet의 順序처럼 생각했을 때 英語 辭典의 單語排列順序와 같게 된다.

以上的 内容을 computer programming한 結果가 역시 附錄에 수록되어 있는 Subroutine SPLP2이며 이의 使用方法도 remarks를 參照하여 주기 바란다.

이 搜索方法을 利用하여 單純 經路를 搜索하는 경우 入力 data에 대한 dimension이 작게 되고 搜索이 신속하게 수행된다는 長點이 있는 반면에 入力 data作成의 번잡성 내지는 그에 대한 變更이 容易하지 못하다는 것과 아울러 應用時에 關聯事項과의 連結이 어렵다는 短點이 있다.

終着點인 node n (여기에서는 node 5)에서 다른 node j ($j \leq n$)로 연결되어 있는 arc는 實際問題를 푸는 過程에서는 不必要한 情報이므로 그림 4와 같은 入力 data(Subroutine SPLP2에서는 NW array에 해당)作成時에는 역시 省略해 버리는 것이 바람직하다.

2. 單純 經路 搜索 프로그램의 應用

앞에서 언급한 單純 經路 搜索의 Subroutine SPLP1 및 SPLP2를 最長 經路의 搜索, 最短 經路의 搜索, network을 통한 最大 流量의 計算, 各 單純 經路의 流量 計算, 信賴度의 計算 등에 應用해 보았으나 여기에서는 最長 經路의 搜索 및 信賴度 計算에 대해서만 언급해 보고자 한다.

(1) 最長 經路의 搜索

대개의 경우 network에 대해서는 단 1개 만의 最長 經路 또는 最短 經路에 대해서 관심을 가지고 이를 搜索해 내는 수가 많으나 때에 따라서는 最長 및 最短 經路의 양쪽에 모두 관심을 갖는 다든가 여러개의 最長 또는 最短 經路에 대해 同時에 관심을 갖는 경우도 있을 것이다. 예를 들면 最長 또는 最短 經路가 어떤 事情에 의해 利用할 수 없게 된 경우에는 그 다음의 最長 또는 最短 經路를 利用할 필요가 생길 것이고, 이것마저도 依의치 못할 경우에는 3번째, 4번째, ……의 最長

또는 最短經路의 利用이 불가피할 것이다.

경우에 따라서는 經路의 길이(距離 또는 時間)가 얼마 이상 또는 얼마 이하이면 곤란하기 때문에 모든 經路들에 대한 길이의 情報가 필요하게 될 수도 있을 것이다.

그래서 위에서 언급한 内容들에 대해

- 1) 주어진 network에 있어서의 特定の 두 node 사이
- 2) 可能的 모든 雙의 nodes 사이
- 3) 위의 1)과 2)에 있어서의 2 번째, 3 번째,

4) 特定の node (s)를 經由하는 경우 등의 最長 및 最短經路의 搜索 및 그 길이(距離 또는 時間)를 計算하는 computer program을 作成하여 假想의 작은 例題에 應用해 보았으나 여기에서는 그 자세한 内容를 省略하고자 한다. 단, 最長短經路의 搜索 및 길이 計算에서는 길이(距離 또는 時間)에 관한 情報의 入力關係로 Subroutine SPLP1을 利用하는 것이 簡便할 것이다.

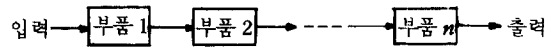
(2) 信賴度 計算

信賴度 計算에서는 時間이 정해졌을 때의 靜信賴度(static reliability)에 대해서만 고려하였고 또한 分析을 容易하게 하기 위해서 network 상의 모든 部品의 作動은 서로 獨立的이라고 가정하였다.

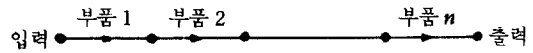
복잡한 system의 信賴度 計算에서는 全體 system을 한꺼번에 다루는 것이 힘들므로 대개의 경우 몇개의 subsystem이나 部品으로 나누어서 생각하는 경우가 많다. 이렇게 system을 細分하면 部品들이 어떻게 相互關聯되어 system을 作動시키는지를 묘사할 수 있는 network을 作成할 수가 있다. 이러한 network은 그림 6과 같은 信賴度 block diagram이나 그림 7과 같은 信賴度 graph로서 흔히 表現하게 되나 여기에서는 單純經路搜索의 subroutine을 쉽게 利用하기 위하여 信賴度 graph로서 system을 network형식으로 表現하는 경우에 대하여 생각하기로 한다.

만일 그림 8과 같은 信賴度 block diagram으로 表現되어 있는 network에 대한 信賴度 計算을 必要로 하는 경우에는 이를 그림 9와 같이 信賴度 graph로 바꾸어 表現한 network에 대하여 計算하면 될 것이다.

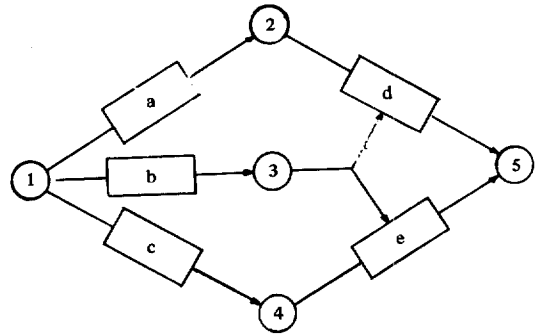
Subsystem의 構造로서는



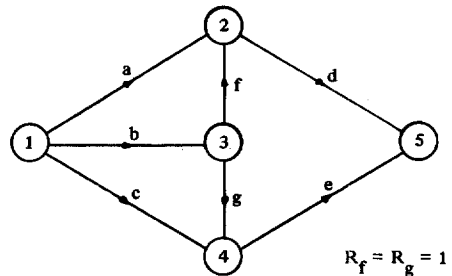
(그림 6) 信賴度 block diagram



(그림 7) 信賴度 graph



(그림 8) 信賴度 block diagram으로 表現한 network의 例



(그림 9) 信賴度 graph로 表現한 network의 例(그림 8과 같은 内容임)

- 直列構造(series structure)
- 並列構造(parallel structure)
- n中 r構造(r-out-of-n structure)

등을 생각할 수 있으나 여기에서는 n 中 r 構造는 省略하기로 한다.

Subsystem이 直列 또는 並列構造로 構成되어 있을 때에는 이를 좀더 簡略化시킬 수 있는 경우도 있다. 이와같은 可略構造(reducible structure)와는 달리 그림 9의 network은 더 이상 簡略化시킬 수 없는 非可略構造(irreducible structure)이다. 여기에서 開發한 computer program은 可略 및 非可略構造가 섞여 있는 信賴度 graph로 表現되어 있는 network의 信賴度計算을 할 수 있도록 programming 되었으나 計算效率를 높이기 위해 可略構造는 일단 非可略構造로 單純化시킨 後 network에 대한 信賴度計算을 하도록 作成되었다.

그리고 program은 TRS-80 Micro Computer System 中 Interface와 Line Printer를 使用하는 것을 前提로 하여 作成되었다.

(1) 事象空間法에 의한 信賴度計算

非可略構造의 信賴度計算에 있어서 가장 單純한 方法이 事象空間法(event-space method)^{11) 10)}이라고 할 수 있을 것이다. 이것은 n 個의 部品이 正常 및 故障狀態인 2가지 경우에 대해 모든 可能的 組合(事象空間의 目錄)을 생각해서 system이 作動되는 組合에서의 信賴度들의 合計를 곧 system全體의 信賴도로 計算하는 方法이다.

이 事象空間法은 部品の 故障이 서로 獨立的이기만 하면 어떠한 system에도 適用할 수 있으며 서로 獨立的이 아니더라도 仔細한 data가 準備만 되어 있으면 使用 가능한 長點이 있다. 그러나 非可略構造로 簡略化시킨 後의 subsystem의 數가 5~6個 以上이면 計算에 많은 時間이 所要되는 短點이 있으나 實際로 그러한 경우는 매우 드물므로 實用上 큰 問題가 되지는 않는다.

前述한 單純經路搜索方法 中 Subroutine SPLP2를 利用하여 computer programming 하였다.

이 program은 micro computer의 記憶容量을 最大로 이용하기 위해 SPLP2를 使用한 關係로 信賴度計算에 必要한 data의 入力이 조금 번거롭게 되어 있다.

Data의 入力에 대한 설명은 program running時 video monitor에 자세히 display 되고 中間計算結果 및 最終結果는 line printer에 의해 出力되도록 하였으므로 여기에서는 省略하고자 한다.

(2) 經路追跡法에 의한 信賴度計算

事象空間法에서는 system이 正常作動할 모든 組合들을 밝혀내야 하며 이에 상당한 時間이 所要된다. 이는 原因과 結果를 連結하는 完全한 經路를 하나하나 밝혀내고 이러한 經路들을 포함하는 事象들의 集合에 대한 確率을 구하는 형식으로 信賴도가 計算되었다. 그러나 system이 正常作動할 完全한 經路들이 처음부터 밝혀진다면 이들의 集合의 確率が 곧 system全體의 信賴도가 될 수 있다.

經路追跡法(path-tracing method)^{11) 10)}은 이와 같은 思考方式下에서 모든 部品이 故障인 경우, 1개의 部品만이 正常인 경우, 2개의 部品이 正常인 경우, ……로 進行해 나가면서 完全經路가 나타나면 멈추어 이들 完全經路에 대한 集合의 確率을 구하게 된다.

經路追跡法은 事象空間目錄을 作成할 필요가 없으므로 事象空間法보다 簡便한 方法이다. 그러나 事象空間法과는 달리 經路追跡法에서는 모든 事象들이 相互排除的이 아니므로 集合의 確率을 구하는 式을 전개하여 計算해 나가는 過程이 매우 복잡하게 되어 computer를 利用하는 경우에는 事象空間法에 비해 별로 우월하다고는 할 수 없다.

經路追跡法을 利用하여 network의 信賴도를 計算하는 computer program 作成에서는 前述한 subroutine SPLP1을 使用하였으며 data의 入力에 대한 설명은 역시 program running時에 video monitor에 자세히 display 되도록 하였고 中間計算結果 및 最終結果와 line printer에 의해 出力되도록 하였으므로 省略한다.

III. 結 論

그 單純性 내지는 素朴한 개념 탓으로 흔히 無視되기 쉬운 單純經路의 應用可能性은 대단히 광범위할 것으로 豫想되어 주어진 network에 있어서의 可能的 모든 單純經路를 搜索하는 方法 2가지에 대해 이를 누구나 쉽게 利用할 수 있도록 BASIC 言語로 subroutine subprogram 형태로 computer programming 하였다. 그리고 이에 대한 一次的인 應用으로서 最長經路의 搜索, 最短短經路의 搜索, network全體의 最大流量의 計算, 單純經路마다의 最大流量計算, 信賴度計算 등에 應用해 보았다.

最長短經路의 搜索에서는 단 1개의 最長 또는 最短短經路 이외에도 2번째, 3번째, 4번째, ……

의 最長 또는 最短經路가 同時에 搜索可能함으로, 단 1개 만의 最長經路를 利用하는 大單位裝備의 維持, 복잡한 工程管理 등의 最長經路 利用分野나, 역시 단 1개 만의 最短經路를 利用하는 裝備의 交替, 投資計劃, 旅路의 決定, set-up 問題 등의 最短經路 利用分野에서 이러한 여러가지 單純經路의 搜索結果는 좀 넓은 용通성을 갖게 할 것이라고 豫想된다. 특히 最近에 國內 電子工業分野 등에서 關心을 보이기 시작한 信賴性工學分野에서 事象空間法 및 經路追跡法에 의한 信賴度計算은 그 實用性이 상당히 높을 것으로 생각된다.

필자가 使用하였던 micro computer의 記憶容量이 48 K로서 비교적 작았던 관계로 program 內에서의 各種 array에 대한 dimension의 크기를 작게 設定했으나 변수 1개(事象空間法에서는 MD, 經路追跡法에서는 DM)의 값만을 바꾸어 주므로서 必要한 dimension의 크기가 自動적으로 調節될 수 있도록 programming 하였으므로 各種 microcomputer의 記憶容量을 최대한으로 活用할 수 있도록 하였다.

本稿에서 언급한 외에도 單純經路의 搜索結果를 應用할 수 있는 分野는 상당히 많을 것으로 豫想하며 좀더 具體적인 應用分野에 대해서는 차후 계속하여 考察해 볼 豫定이다.

參 考 文 獻

- (1) 朴 景洙, 信賴性工學 및 整備理論, 塔出版社, 1978.
- (2) 鄭秀一, “最長經路 Algorithm의 計算 能率比較”, 大韓產業工學會誌, Vol. 1, No. 1, June 1975
- (3) 鄭秀一, “最長經路 Algorithm의 計算能率比較(其二)”, 仁荷大學校產業科學技術研究所 論文集, 第5輯, pp. 63-66, 1978年 5月
- (4) 鄭秀一, “最長經路 Algorithm의 考察 및 그 能率比較(綜合)”, 仁荷大學校產業科學技術研究所 論文集, 第6輯, pp. 23-27, 1979年 5月
- (5) 鄭秀一, “단순경로수색의 프로그래밍 및 그 응용에 관한 연구(I)”, 仁荷大學校產業科學技術研究所 論文集, 第8輯, pp. 77-83, 1981年 4月
- (6) 鄭秀一, “단순경로수색의 프로그래밍 및 그 응용에 관한 연구(II)”, 仁荷大學校產業科學技術研究所 論文集, 第10輯, pp. 53-57, 1982年 6月
- (7) 卓永奉, “計算機프로그램에 의한 通信回路網의 單純通路決定”, 仁荷大學校 大學院 電子工學科, 工學碩士學位論文, 1979年 2月
- (8) Barlow, Richard E. and Proschan, Frank, *Mathematical Theory of Reliability*, John Wiley & Sons, Inc., New York, 1965.
- (9) Kroft, D., “All Paths Through A Maze”, Proc. IEEE (Lett.), Vol. 55, pp. 88-90, Jan. 1967.
- (10) Shooman, Martin L., *Probabilistic Reliability; an engineering approach*, McGraw-Hill, New York, 1968.

```

9000 REM
9001 REM *****
9002 REM *          SUBROUTINE SPLP1          *
9003 REM *          FOR SEARCHING AND IDENTIFYING          *
9004 REM *          ALL THE SIMPLE PATHS          *
9005 REM *          FROM NODE 1 TO NODE N IN A GIVEN NETWORK          *
9006 REM *****
9007 REM
9008 REM          DESCRIPTION OF THE INPUT ARGUMENTS
9009 REM          =====
9010 REM N ..... NUMBER OF NODES ..... VARIABLE
9011 REM FT ..... FROM-TO TABLE ..... VECTORED ARRAY
9012 REM FT(I,J) > 0 FOR RELIABLE ARC (I,J)
9013 REM FT(I,J) <= 0 FOR UNRELIABLE ARC (I,J)
9014 REM
9015 REM          DESCRIPTION OF THE OUTPUT ARGUMENTS
9016 REM          =====
9017 REM PN ..... TOTAL NUMBER OF PATHS ..... VARIABLE
9018 REM AC ..... NETWORK CHARACTERISTIC ..... VARIABLE
9019 REM AC = 0 FOR ACYCLIC NETWORK
9020 REM AC > 0 FOR CYCLIC NETWORK
9021 REM EP ..... NUMBER OF NODES IN EACH PATH ..... ARRAY
9022 REM JN ..... J-TH NODE OF I-TH PATH ..... VECTORED ARRAY
9023 REM
9024 REM          NECESSARY DIMENSION FOR SPLP1
9025 REM          =====
9026 REM DIM CJ(5*N), EP(10*N), FR(N, 5*N), FT(N, N), IP(N, 5*N),
9027 REM      JN(10*N, N), MO(N, 5*N), SP(N)
9028 REM
9029 REM          RESERVED VARIABLES FOR SPLP1
9030 REM          =====
9031 REM AC, FR, IM, IO, IT, JH, JM, KR, N, NT, PN, PO
9032 REM
9033 IT=1: CJ(1)=1: IP(1, 1)=1: FR(1, 1)=0: MO(1, 1)=0: PN=0: AC=0
9034 PO=IT+1: CJ(PO)=0: FOR I=1 TO CJ(IT): FR=IP(IT, I): FOR J=1 TO N
9035 IFFT(FR, J)>0 THEN 8036 ELSE 8046
9036 IF J=N THEN 8037 ELSE 8041
9037 PN=PN+1: SP(PO)=N: SP(IT)=IP(IT, I): IM=FR(IT, I): JM=MO(IT, I)
9038 IO=IT-1: FOR K=1 TO IO: KR=IT-K: SP(KR)=IP(IM, JM): IH=FR(IM, JM)
9039 JH=MO(IM, JM): IH=IH: JM=JH: NEXT K: EP(PN)=PO: FOR L=1 TO PO
9040 JN(PN, L)=SP(L): NEXT L: GOTO 8047
9041 IM=FR(IT, I): JM=MO(IT, I): IO=IT-1: FOR K=1 TO IO: KR=IT-K
9042 SP(KR)=IP(IM, JM): IH=FR(IM, JM): JH=MO(IM, JM): IH=IH: JM=JH
9043 NEXT K: FOR L=1 TO IO: IF J=SP(L) THEN 8046
9044 NEXT L: CJ(PO)-CJ(PO)+1: NT=CJ(PO): IP(PO, NT)=J: FR(PO, NT)=IT
9045 MO(PO, NT)=I: GOTO 8047
9046 AC=AC+1
9047 NEXT J, I: IFCJ(PO)>0 THEN IT=IT+1: GOTO 8003
9048 RETURN

```

```

9000 REM
9001 REM *****
9002 REM *          SUBROUTINE SPLP2          *
9003 REM *          FOR SEARCHING AND IDENTIFYING          *
9004 REM *          ALL THE SIMPLE PATHS          *
9005 REM *          FROM NODE 1 TO NODE N IN A GIVEN NETWORK          *
9006 REM *****
9007 REM
9008 REM          DESCRIPTION OF THE INPUT ARGUMENTS
9009 REM          =====
9010 REM NN ..... NUMBER OF NODES ..... VARIABLE
9011 REM ND ..... NUMBER OF NETWORK DATA(NM) ..... VARIABLE
9012 REM NW ..... NETWORK INFORMATION ..... ARRAY
9013 REM          IN THE FOLLOWING FORM, FOR EXAMPLE,
9014 REM          NW(1)      NW(5)
9015 REM          . NW(2)      NW(6)
9016 REM          . . NW(3)      NW(7)
9017 REM          . . . . .
9018 REM          . . . . .
9019 REM          . . . . .
9020 REM          -1, 2, 3, 4, -2, 1, 5, 7, 9, -3, 2, .....
9021 REM          . . . . .
9022 REM          . . . . .
9023 REM          . . . . .
9024 REM          . . . . . TO . . . . . TO
9025 REM          . . . . . TO . . . . . TO
9026 REM          FROM . . . . . FROM
9027 REM
9028 REM          DESCRIPTION OF THE OUTPUT ARGUMENTS
9029 REM          =====
9030 REM TP ..... TOTAL NUMBER OF PATHS ..... VARIABLE
9031 REM PN ..... NUMBER OF NODES IN EACH PATH ..... ARRAY
9032 REM NJ ..... J-TH NODE OF I-TH PATH ..... VECTORED ARRAY
9033 REM
9034 REM          NECESSARY DIMENSION FOR SPLP2
9035 REM          =====
9036 REM DIM CH(NN), NJ(15*NN, NN), NW(6*NN+1), PN(15*NN), PT(NN)
9037 REM
9038 REM          RESERVED VARIABLES FOR SPLP2
9039 REM          =====
9040 REM IN, JN, KN, LN, ND, NN, TP, TT
9041 REM
9042 TP=0: IF NW(1)=-1 THEN NN=(ND+1)=0 ELSE RETURN
9043 FOR I=1 TO NN: CH(I)=0: NEXT I: IN=1: JN=1: PT(1)=1
9044 KN=ABS(NW(IN)): CH(KN)=1: LN=1: GOTO 9048
9045 IF NW(LN)=0 THEN 9058 ELSE 9046
9046 IF NW(LN)<0 THEN 9048 ELSE 9047
9047 LN=LN+1: GOTO 9045
9048 IF ABS(NW(LN))=ABS(NW(IN)) THEN 9058 ELSE 9049
9049 IF NW(LN)<0 THEN 9047 ELSE 9045
9050 LN=LN+1: IN=LN: JN=JN+1
9051 TT=NW(IN): IFCN(TT)=0 THEN 9052 ELSE 9055
9052 PT(JN)=IN: IF NW(IN)=N THEN 9053 ELSE 9044
9053 TP=TP+1: PN(TP)=JN
9054 FOR I=1 TO JN: TT=PT(I): NJ(TP, I)=ABS(NW(TT)): NEXT I
9055 TT=IN+1: IF NW(TT)>0 THEN 9056 ELSE 9057
9056 IN=IN+1: GOTO 9051
9057 JN=JN-1: IN=PT(JN): KN=ABS(NW(IN)): CH(KN)=0
9058 IF NW(IN)=-1 THEN RETURN ELSE 9055

```