韓國經營科學會誌
第23卷 第4號
1998年 12月

최단경로문제에서 k개의 치명호를 찾는 방법*

안재근** · 정호연*** · 박순달****

A Method for Finding the k Most Vital Arcs in the Shortest Path Problem*

Jae-geun Ahn** · Ho-yeon Chung*** · Soon-dal Park****

■ Abstract ■

This paper deals with a mathematical model and an algorithm for the problem of determining k most vital arcs in the shortest path problem.

First, we propose a 0-1 integer programming model for finding k most vital arcs in shortest path problem given the ordered set of paths with cardinality q. Next, we also propose an algorithm for finding k most vital arcs in the shortest path problem which uses the 0-1 integer programming model and shortest path algorithm and maximum flow algorithms repeatedly.

Malik et al. proposed a non-polynomial algorithm to solve the problem, but their algorithm was contradicted by Bar-Noy et al. with a counter example to the algorithm in 1995. But using our algorithm, the exact solution can be found differently from the algorithm of Malik et al.

1. 서 론

최단경로문제는 특정한 두 교점 사이의 경로 중에서 가장 짧은 길이의 경로를 찾는 문제이다[1, 3]. 이 최단경로문제에서 하나의 호나 여러 개의 호가 동시에 제거된다고 가정하면 제거된 호에 따

* 본 연구는 정보통신부 '97년도 대학기초연구지원사업(97-G-0683)의 지원을 받음.
** 안성산업대학교 컴퓨터공학과
*** 전주대학교 산업공학과
**** 서울대학교 산업공학과

라 최단경로의 길이가 늘어날 수 있게 된다. 이 때 한 호의 제거로 최단거리가 가장 크게 늘어나는 호를 최단경로문제에서의 치명호(Most Vital Arc : 1-MVA)라 하고, k개의 호의 제거로 시발점에서 종착점까지의 최단거리가 가장 크게 늘어나는 k개의 호집합을 찾는 문제를 k개의 치명호(k Most Vital Arcs : k-MVA)를 찾는 문제라고 한다[2, 7]. 이러한 연구는 적의 공격 하에 처해 있는 이해상충의 상황이나 물류 또는 통신네트워크에서 어느 호가 가장 치명적인지 알아내어 적의 공격으로부터 경계를 강화하거나 어느 호를 파괴시켜야 적의 시스템의 효율성을 가장 크게 저하시킬 수 있는지 알고자 하는 문제에 잘 적용될 수 있다[6]. 따라서 군사용 보급로나 물류에서의 수배송망 관리, 통신망, 송수신 네트워크 등 호의 제거가 전체 네트워크의 성능에 치명적으로 영향을 미치는 경우의 네트워크문제에 활용될 수 있다.

이에 대한 연구는 Corley와 Sha[5], Malik 외 2인[7], Bar-Noy 외 2인[4]에 의해서 수행되었다. Corley와 Sha[5]는 k번째 최단경로(k-shortest path)를 구하는 해법을 이용하여, 최단경로문제에서 1개의 MVA(1-MVA)를 찾는 해법과, k개의 MVA 찾는 문제(k-MVA)에 대한 충분조건을 제시하였다. Malik 외 2인[7]은 Corley와 Sha의 방법을 개선하여 무방향 네트워크에서 1-MVA를 찾는 해법을 제시하였고, k-MVA에 대해서는 k개의 MVA를 찾는 비다항시간의 복잡도(non-polynomial time complexity)를 가지는 해법을 제시하였다. 그러나 이 문제는 Bar-Noy 외 2인 [4]에 의해 반례가 제시됨으로써 해법의 오류가 있음이 판명됨과 동시에 k개의 치명호를 찾는 문제가 NP-hard임이 증명되었다.

따라서, 본 연구는 Corley와 Sha의 k-MVA가 가지는 충분조건과 Malik 외 2인이 제시한 다수번째의 최단경로의 해법을 이용하여 기존의 방법으로 최적해를 제공하지 못하는 최단경로문제에서의 k개의 치명호를 찾는 문제의 수리모형 및 알고리즘을 개발 그리고 해법을 효율화 할 수 있는 방법에 대해 다루는 것을 목적으로 한다.

2. 기본용어 정의

본 연구에서 다루는 최단경로문제 $G=(N,A)$는 마디 수가 |N|=n 이고, 호의 수가 |A|=m 인 비음의 호 길이(non-negative arc length)를 갖는 유방향(directed) 가중 네트워크라 가정한다.

본 논문에서 사용하는 주요 용어를 정리하면 다음과 같다.

- s, t : $G=(N,A)$의 시발점(source node)과 종착점(sink node).
- d_{ij} : 호 $(i,j)\in A$의 길이.
- k ($k\le m$, $k\ge 1$인 정수) : 구하고자 하는 치명호의 갯수.
- L_k : 임의의 k개의 호로 구성된 호집합 ($L_k \subseteq A$).
- L_k^* : k개의 호로 구성된 k-MVA 집합 ($L_k^* \subseteq A$).
- r_j : 시발점과 종착점간의 j번째 최단경로.
- $(s,b_1)(a_2,b_2)\cdots(a_{p_j},t)$, $\forall(s,b_1),\cdots,(a_{p_j},t) \in A$: $s\rightarrow t$까지의 경로를 구성하고 있는 호의 순서쌍(단, p_j는 r_j를 이루는 호의 갯수).
- $d(r_j)$: $s\rightarrow t$까지의 j번째 최단경로 r_j의 길이. 편의상 $d(r_j)$를 d_j로 나타내기도 함.
- $s\rightarrow t$까지의 j번째 최단경로 r_j와 $s\rightarrow t$까지의 (j+1)번째 최단경로 r_{j+1}사이에는 $d(r_j)$ $\le$ $d(r_{j+1})$가 성립한다고 가정.
- $R=\{r_1, r_2\cdots, r_q\}$: $s\rightarrow t$까지의 모든 경로들의 집합(단, q는 $s\rightarrow t$까지의 경로의 최대 갯수).
- q_1 : 호 k개를 제거했을 때 길이의 오름차순(ascending order)으로 정렬된 최단경로를 순서대로 절단시키는 절단경로의 최대갯수.

3. k개의 치명호를 찾는 수리모형

최단경로문제의 k-MVA를 찾는 문제에 대하여 기존의 연구에서 밝혀진 사실은 k-MVA를 찾는 문제가 NP-hard[4]라는 것과 k-MVA는 1-MVA를 구하는 해법을 순차적으로 k회 반복 적용하여 얻을 수 없다는 것이다[7].

k-MVA를 찾는 문제가 NP-hard라는 사실은 Bar-Noy 외 2인에 의하여 증명되었는데, 이들은 Node Cover Decision 문제가 단위 길이의 호를 갖는 네트워크에서의 치명호를 결정하는 의사결정문제로 전환(reduce)됨을 보임으로써 k-MVA를 찾는 문제가 NP-hard임을 증명하였다[4]. k-MVA가 1-MVA를 찾는 해법을 순차적으로 k회 반복 적용하여 구한 것과 같지 않다는 것은 k-MVA가 단순히 1-MVA 찾는 문제를 반복 적용하여 구할 수 없다는 사실을 말해 준다[7].

Corley와 Sha[5]는 k-MVA에 대한 충분조건을 제시하여 $s \to t$를 연결하는 모든 경로가 k개의 호로 모두 절단될 수 있거나, $s \to t$를 연결하는 최단경로의 수가 k개 보다 큰 경우에 대하여 k-MVA를 찾는 방법을 제시하였다.

따라서 본 연구에서는 $s \to t$를 연결하는 모든 경로가 k개의 호로 모두 절단될 수 없고, 또한 최단경로의 수가 k개 보다 작은 경우에 k-MVA를 찾는 방법을 제시해 보고자 한다.

본 연구에서 k-MVA를 찾는 기본 개념은 다음과 같다. k-MVA를 찾기 위해 임의의 k개의 호의 제거를 통해 최대 q_1개의 경로를 길이의 오름차순에 따라 절단할 수 있는 k개로 구성된 호집합과 절단하지 못하는 k개로 구성된 호집합이 있을 경우, 후자의 경우에 k개 호의 제거를 통해 r_{q_1}의 길이 만큼은 늘려줄 수 없기 때문에 전자의 k의 호가 후자의 경우보다 더 치명적(vital)이게 된다는 사실을 이용한다.

L_k 또는 L_k^*는 제거되는 호를 의미한다. 그러므로, 네트워크 $G=(N,A)$에서 k개의 호로 구성된 특정 L_k를 제외한 네트워크 $G=(N,A \backslash L_k)$에서의 최단경로의 길이는 k개의 호를 제거되기 전보다 크거나 같을 수 있다. 그리고 특정 $G=(N,A \backslash L_k)$에서의 최단경로는 $s \to t$까지의 경로 중에서 그 길이가 가장 짧은 경로이다. 즉, $G=(N,A \backslash L_k)$에서의 최단경로는 $G=(N,A)$에서 L_k가 제거되었을 때의 최단경로의 길이로 표현할 수 있다. 그런데 k-MVA는 정의에 의해 k개의 호의 제거로 시발점에서 종착점까지의 최단거리가 가장 크게 늘어나는 k개의 호집합이다. 그러므로 k-MVA L_k^*가 제거된 네트워크 $G=(N,A \backslash L_k^*)$는 L_k가 제거된 $G=(N,A \backslash L_k)$들 중에서 가장 최단경로의 길이가 큰 k개의 호로 구성된 호집합이다. 그러므로 k-MVA를 찾는 문제를 경로와 호의 관계로 표현하여 정리하면 [성질 1]과 같다.

[성질 1] 네트워크 $G=(N,A)$의 최단경로문제에서 k-MVA를 찾는 문제는, $G=(N,A \backslash L_k)$에서 L_k를 구성하는 호가 하나도 포함되지 않는 $s \to t$까지의 경로 중에서 그 길이가 가장 짧은 경로를 가장 크게 하는, k개의 호로 구성된 $L_k^* (\subseteq L_k)$를 찾는 문제이다.

이는 $G=(N,A)$에서의 k-MVA를 찾는 문제가 k개의 호로 이루어진 모든 호집합 L_k를 제외한 네트워크 $G=(N,A \backslash L_k)$들 중에서 최단거리가 가장 긴 $L_k^* \subseteq L_k$를 찾는 문제와 같다는 사실을 말해준다[5].

그러나 [성질 1]을 사용하여 k-MVA를 구하기 위해서는 호집합 L_k 각각이 제거된 상태에서의 최단경로를 구할 수 있는 효율적인 방법이 존재하

여야 한다. 그러나 일반적으로 m개의 호에서 k개를 구하는 방법은 조합의 개수만큼 필요하므로 위의 [성질 1]을 이용하기가 쉽지 않다. 그러므로 경로와 호의 관계로 표현된 [성질 1] 대신 k-MVA를 찾는 문제를 다수번째 최단경로문제(k-shortest path problem)의 해법을 사용할 수 있는 형태로 표현하면 다음과 같다.

[성질 2] 네트워크 $G=(N, A)$의 경로길이의 오름차순으로 정렬된 모든 경로들을 알고있다고 가정하자. 그러면, k-MVA를 찾는 문제는, 호집합 L_k를 구성하고 있는 호들이 적어도 하나 이상 포함되는 경로들이, 첫 번째 최단경로로부터 짧은 길이의 경로를 빠뜨리지 않고 오름차순으로 가장 많은 경로들을 포함하는 호집합 $L_k^*(\subseteq L_k)$를 찾는 문제이다.

그러나 다음과 같은 관찰에 의해 구하고자 하는 k-MVA모형을 변형하여 치명호에 속한 호의 제거를 통해서 절단되는 경로 중에서 치명호 결정에서 고려하지 않아도 되는 경로들을 목적함수에서 제외하는 모형을 구하고자 한다.

[성질 3] 네트워크 $G=(N, A)$에서 k개의 치명호 L_k^*를 제거한 후의 $G=(N, A \backslash L_k^*)$에서의 최단경로의 길이를 L이라고 가정하자. 그러면 치명호 집합 L_k^*을 제거하기 이전의 네트워크 $G=(N, A)$에서 경로의 길이가 L보다 짧은 경로들 ($d(r_j) < L$)은 모두 L_k^*를 구성하는 호를 적어도 하나 이상 가진다. 그리고 치명호 집합 L_k^*을 제거하기 이전의 네트워크 $G=(N, A)$에서 경로의 길이가 L인 경로($d(r_j) = L$)에는 L_k^*를 구성하는 호를 가지고 있지 않다.

따라서 [성질 3]에 의해 L_k^*의 제거를 통해 경로의 길이가 L 이상인 경로들에 대해서는 치명호가 포함되지 않기 때문에 k-MVA 집합을 구하는데 필요한 경로에서 제외시켜도 무방하다.

이상의 성질들을 이용하여 k-MVA를 결정하는 수리 모형식을 만들기 위한 변수를 정의해 보자. 먼저 치명호를 나타내는 결정변수 x_i는 호 i가 치명호집합에 속하면 1, 그렇지 않으면 0을 갖는 결정변수라 하고, z_j는 치명호의 제거를 통해 j번째 경로가 $s \to t$까지의 연결이 절단되면 1, 그렇지 않으면 0을 갖는 변수이며, y_{ij}는 호 i가 j번째 경로에 속하면 1, 그렇지 않으면 0을 갖는 변수라고 정의하자. 이 때 q개의 경로 $r_1, \cdots, r_q$가 순차적으로 주어졌을 때 L_k가 치명호가 되기 위해서는 L_k에 속한 호의 제거가 경로길이에 대한 오름차순으로 나열된 경로를 순차적으로 제거시킬 수 있어야 한다. 만일 호 i가 치명호 집합에 속해서 제거된다면(즉, $x_i=1$), 호 i가 j번째 경로의 구성요소인지 여부를 나타내는 변수인 y_{ij}에 의해, j번째 경로가 절단될 수 있음($z_j=1$)을 수리식으로 나타낼 수 있다. 이를 수식으로 표현하면 [보조정리 1]과 같다.

[보조정리 1] q개의 경로 $r_1, \cdots, r_q$가 순차적으로 주어졌을 때 $d_j \leq d_{j+1}$인 경우 $z_j=1$값을 갖기 전에는 $z_{j+1}=1$이 될 수 없게 하는 수식은

$$d_{j+1} \cdot z_j - d_j \cdot z_{j+1} \geq 0, \quad \forall\, j=1, \cdots, q-1, \quad \text{단, } z_j \in \{0, 1\}, \ \forall\, j=1, \cdots, q \tag{a}$$

이다. (여기서 d_j는 $d(r_j)$를 나타내는 수식이다)

(증명)

$d_{j+1} \geq d_j$ 인 경우 변수 z_j가 1이 되기 이전에 변수 z_{j+1}이 1을 가질 수 없음을 보이기 위해서는 z_j와 z_{j+1} 의 관계에서 $z_j = 0$ 일 때 $z_{j+1}=1$인 경우가 성립되지 않게 하는 수식을 만들면 된다.

따라서 주어진 수식 $d_{j+1} \cdot z_j - d_j \cdot z_{j+1} \geq 0$, $\forall j=1,\cdots,q-1$ 을 분석해 보면 $d_{j+1} \geq d_j$ 일 때 만일 $z_j = z_{j+1} = 1$ 인 경우나 $z_j = 1$, $z_{j+1} = 0$ 또는 $z_j = z_{j+1} = 0$ 인 경우는 주어진 부등식이 성립하지만 $z_j = 0$, $z_{j+1} = 1$ 인 경우는 주어진 부등식이 성립하지 않는다. 따라서 $d_{j+1} \cdot z_j - d_j \cdot z_{j+1} \geq 0$, $\forall j=1,\cdots,q-1$식은 경로에 해당하는 변수 $z_1,\cdots,z_q$가 경로의 길이가 짧은 변수가 1이 되기 이전에 길이가 긴 변수는 1을 가질 수 없는 경우에 해당하는 수식이 된다. □

구하고자 하는 치명호의 갯수가 k개임을 표현하는 수식은 다음과 같이 식 (b)로 나타낼 수 있다.

$$\sum_{i \in A} x_i = k \qquad \text{(b)}$$
$$x_i \in \{0,1\},\ i=1,\cdots,m$$

호 i가 $s \to t$ 간의 경로의 구성요소임을 표현해 주는 수식은 다음의 식(c)와 같다.

$$x_i - y_{ij} = 0, j=1,\cdots,q,\ \forall i \in r_j \qquad \text{(c)}$$
$$\sum_{i \in r_j} y_{ij} - z_j \geq 0, j=1,\cdots,q$$

$$x_i \in \{0,1\},\ i=1,\cdots,m$$
$$y_{ij} \in \{0,1\},\ j=1,\cdots,q,\ \forall i \in r_j$$
$$z_j \in \{0,1\},\ j=1,\cdots,q$$

위의 [보조정리 1]의 식(a)와 치명호의 갯수가 k개임을 나타내는 식(b) 그리고 호가 경로의 구성요소임을 표현해 주는 식(c)를 통해 짧은 경로 순서대로 순차적으로 경로를 절단할 수 있는 k개의 호를 구하는 방법을 수식으로 표현할 수 있다. 또한 [성질 2]에서 L_k를 구성하고 있는 호가 적어도 하나 이상 포함되는 $s \to t$ 까지의 경로들을 경로길이에 대한 오름차순으로 정렬했을 때, 최단경로로부터 순차적으로 가장 많은 경로에 포함되는 k개의 호 집합을 찾는 문제는 호의 제거를 통해 j번째 경로가 $s \to t$ 까지 연결되었는지 여부를 나타내는 변수인 z_j을 최대한 많이 선택하게 함으로써 얻을 수 있다.

이를 수식으로 나타내면 식 (d)와 같다.

$$\max \sum_{r_i \in R} z_j \qquad \text{(d)}$$

따라서 최단경로문제의 k-MVA를 찾는 문제에 대한 수리모형식은 식 (d)를 목적함수로 놓고 식 (a)-(c)를 제약식과 변수제약으로 놓고 풀면 된다. 이를 정리한 모형식이 다음의 k-MVASP 이다.

(k-MVASP)

$$\max \sum_{r_i \in R} z_j \qquad (1)$$
$$s.t. \sum_{i \in A} x_i = k \qquad (2)$$
$$x_i - y_{ij} = 0, j=1,\cdots,q,\ \forall i \in r_j \qquad (3)$$
$$\sum_{i \in r_j} y_{ij} - z_j \geq 0, j=1,\cdots,q \qquad (4)$$
$$d_j z_l - d_l z_j \geq 0, l=1,\cdots,q,\ j=l+1,\cdots,q$$
$$x_i \in \{0,1\},\ i=1,\cdots,m \qquad (5)$$
$$y_{ij} \in \{0,1\},\ j=1,\cdots,q,\ \forall i \in r_j$$
$$z_j \in \{0,1\},\ j=1,\cdots,q$$

4. k개의 치명호문제의 해법의 효율화

(k-MVASP) 모형을 사용하기 위해서는 사전에 $s \to t$간의 모든 경로를 알아야 한다. 그러나 $s \to t$ 간의 모든 경로를 구한다는 것은 일반적으로 마디나 호의 수가 증가함에 따라 경로의 수가 지수적으로 증가하는 성질 때문에 바람직하지 않다. 따라서 $G=(N, A)$의 $s \to t$를 연결하는 모든 경로가 k개 또는 k개 보다 작은 수의 호로 모두 절단될 수 있는 경우나 $G=(N, A)$의 $s \to t$간의 서로 교차하지 않는 첫번째 최단경로의 수가 k보다 클 경우에는 다음의 [보조정리 2]를 이용하면 좀 더 간단한 방법으로 구할 수 있다[7].

[보조정리 2] $s \to t$간의 서로 교차하지 않는 경로

의 최대갯수는 주어진 네트워크에서 호들을 제거해서 $s \rightarrow t$간의 모든 경로를 절단시키는 최소의 호의 갯수와 같다[2].

$s \rightarrow t$간의 서로 교차하지 않는 경로의 최대갯수는 주어진 $G=(N, A)$의 모든 호의 용량상한을 1로 놓은 다음 단위용량 최대유통문제를 풀어서 구할 수 있다. 이렇게 하여 구한 최대유통량(m_1)과 제시된 k의 값에 따라 k-MVA를 결정하는 방법은 다음과 같다.

만일 m_1이 k보다 작으면, k-MVA는 단위용량 최대유통문제의 최소절단에 속한 호들과 이 호들을 제외한 (k - m_1)개의 임의의 호로 구성된다. 만일 m_1이 k와 같으면 단위용량최대유통문제의 최소절단에 해당하는 k개의 호들이 k-MVA가 된다. 만일 m_1이 k보다 클 경우에는 $s \rightarrow t$간의 모든 첫 번째 최단경로들을 구한 다음, 이 첫 번째 최단경로들에 속하는 모든 호들의 용량상한은 1로, 그리고 최단경로에 속하지 않는 호들의 용량상한은 0으로 놓고 단위용량최대유통문제를 푼다. 이 때의 최대유통량이 k보다 크면 k-MVA는 임의의 k개 호가 된다. 그러나 만일 최대유통량이 k보다 작으면 (k-MVASP)의 수리모형식을 사용하여 k-MVA를 구해야 한다. 그러나 이 경우에 있어서도 다음과 같은 성질 때문에 실제 $s \rightarrow t$간의 모든 경로를 구할 필요 없이 수개의 최단경로를 나열해서 (k-MVASP)를 적용하고, 이때의 목적함수값이 나열된 경로만큼 나올 경우에 한해 다시 추가적으로 다수최단경로문제의 결과를 이용하는 방법을 사용함으로써 계산횟수를 줄일 수 있다.

[성질 4] 만일 p($1 \leq p \leq q$, 정수)개의 오름차순으로 정렬된 경로를 구하여 (k-MVASP) 모형을 풀었을 경우 다음과 같은 세가지 경우가 생길 수 있다. 단 q는 $s \rightarrow t$간의 경로의 최대갯수를 의미한다.

첫째, (k-MVASP)의 최적해의 목적함수값이 p일 경우(단, p < q)이다. 이 경우는 k개의 호를 제거하여 p개 이상의 경로를 절단할 수 있는 경우에 해당되므로 현재의 p개의 경로만으로 k-MVA를 구할 수 없다. 그러므로 추가적인 경로를 구해서 (즉, p를 증가시켜서) 다시 (k-MVASP)를 적용할 필요가 있다.

둘째, (k-MVASP)의 최적해의 목적함수값이 p보다 작을 경우(단, p < q)이다. 이 경우는 k개의 호를 제거하여 현재 나열된 p개의 경로 중에서 목적함수값 만큼의 경로를 절단할 수 있다. 따라서 현재의 모형의 해 x_i 중에서 비영의 값을 가지는 호가 다수치명호가 된다.

셋째, p = q인 경우이다. 이 경우에는 더 이상의 경로를 나열 할 수 없으므로 모형(k-MVASP)의 해 x_i 중에서 비영의 값을 가지는 호가 다수치명호가 된다.

[성질 4]는 $s \rightarrow t$간의 경로를 필요할 때 순차적으로 구해서 모형에 적용하는 경우가 모든 경로를 나열한 다음에 해법을 적용하는 것보다 유리하다는 사실을 말해 준다. 만일 p-1 개의 경로가 k개의 호의 제거로 절단이 가능하다고 알려졌다면 다음으로 p 번째 경로가 k개의 호의 제거로 절단될 수 있는가를 알 필요가 있다. 이를 알면 최단경로로부터 경로길이의 오름차순으로 순차적으로 경로를 하나씩 구한 다음, 구하여진 경로가 k개의 호의 제거로 절단되는지를 검토해서 추가적으로 경로를 나열할 것인지 끝낼 것인지를 결정할 수 있기 때문이다.

이러한 특성을 반영한 모형식이 (k-MVASP-p)이다. 즉, (k-MVASP-p) 모형식은 (k-MVASP)의 제약식 (4)를 기존에 절단될 수 있었던 p-1개의 경로들에 대한 제약식 (4-1) 과 p 번째 경로의 절단여부를 나타내는 제약식 (4-2)로 변형하였다. 또한 (k-MVASP)의 제약식 (5)는 모형식 (k-MVASP-p)의 제약식 (5-1)로 나타내었고, y_{ij}는

(k-MVASP)모형에서도 정수변수로 처리할 필요가 없는 변수이기 때문에 실수로 처리할 수 있다.

이를 정리하면 다음과 같다.

(k-MVASP-p)

$$\max \ z_p \quad (1)$$

$$s.t. \ \sum_{i \in A} x_i = k \quad (2)$$

$$x_i - y_{ij} = 0, \ j=1,\cdots,p, \ \forall \, i \in r_j \quad (3)$$

$$\sum_{i \in r_j} y_{ij} \geq 1, j=1,\cdots,p-1 \quad (4\text{-}1)$$

$$\sum_{i \in r_p} y_{ip} - z_p \geq 0 \quad (4\text{-}2)$$

$$1 - z_p \geq 0 \quad (5\text{-}1)$$

$$x_i \in \{0,1\}, \ i=1,\cdots,m$$

$$y_{ij} \geq 0, \ j=1,\cdots,p, \ \forall \, i \in r_j$$

$$z_j \in \{0,1\}, \ j=1,\cdots,p$$

위의 모형식을 통해 k나 k이하 갯수의 호의 제거로 이미 p-1개의 경로가 절단되었을 때 p 번째의 경로가 k개 호의 제거로 절단이 가능한지를 알 수 있다. 이는 (k-MVASP-(p-1)) 모형식의 목적함수의 값이 1 이었음을 의미한다. 그러므로 p개의 경로를 구한 다음 이를 (k-MVASP)에 적용하여 구한 해와 (k-MVASP-p)을 이용하여 구한 해는 동일하게 된다.

이를 정리하면 다음과 같다.

[성질 5] 만일 k개의 호의 제거로 p-1 개의 경로가 절단이 가능하고, p($1 \leq p \leq q$,정수)번째의 경로를 구하여 (k-MVASP-p) 모형에 적용한 결과, 최적해의 목적함수값이 1일 경우에는 현재의 k개의 호의 제거로 p개 이상의 경로길이에 대한 오름차순으로 정렬된 경로들을 제거할 수 있다. 그렇지 않고 최적해의 목적함수값이 0이 나올 경우에는 p-1 개의 경로를 절단하는 k개의 호집합이 치명호집합이 된다.

위 (k-MVASP-p)모형에서 최적해의 목적함수가 1이 나올 경우 [성질 4]와 같이 다시 하나의 최단경로를 추가적으로 관찰할 필요가 있다. 만일 목적함수값이 0이 나올 경우에는 현 단계에서 구한 x_i가 최단경로문제의 다수치명호가 된다. 그런데 모형식 (k-MVASP)은 (호의 갯수 + 경로의 갯수)의 0-1 정수변수로 표현 됨에 반하여 모형식 (k-MVASP-p)은 (호의 갯수 + 1)개의 0-1정수변수로 표현된다. 그러므로 임의의 p개의 경로를 구하여 위의 수리모형을 푼 결과가 목적함수값이 1 이 나올 경우 p 를 증가시켜서 수리모형을 반복 적용할 필요가 있다. 이 [성질 5]를 이용하면 모든 경로를 나열할 필요 없이 다수최단경로문제의 해법을 통해 얻어진 경로길이의 오름차순으로 생성된 경로를 이용하여 반복적으로 수리모형을 적용하여 해를 구할 수 있게 된다.

지금까지의 내용을 정리하면 다음과 같다.

k개의 치명호 계산 알고리즘

단계 1 (초기화)

$G=(N,A)$에서 용량상한을 1로 둔다($u_{ij}=1$, $\forall (i,j) \in A$).

단위용량최대유통문제를 푼다.

만일 단위용량최대유통문제의 최대유통량(m_1) < k 보다 작을 경우 다수치명호는 m_1개의 단위용량최대유통문제의 최소절단에 속한 호들과 이 호를 제외한 임의의 호들 중에서 ($k - m_1$)개의 호로 구성된다. 끝.

만일 $m_1 = k$ 이면 단위용량최대유통문제의 최소절단에 해당하는 k개의 호가 다수치명호를 형성한다. 끝.

아니면 p = k-1, p = 1로 놓는다. 단계 2로 간다.

단계 2 (자명해 여부 판정)

p번째의 최단경로를 구한다.

만일 p번째 최단경로가 존재하지 않을 경우 단계 4로 간다.

만일 p 번째 최단경로의 길이가 1번째 최단경로의 길이와 같을 경우

1번째 최단경로와 길이가 같은 모든 경로를 구한다.

이 경로에 포함되는 모든 호에 대해 1의 용량상한, 이외의 모든 호에 대해 0의 용량상한을 두고 단위용량최대유통문제를 푼다.

이때 최대유통량이 k보다 크면 다수치명호는 주어진 네트워크의 임의의 k개의 호가 된다. 끝.

아니면 구한 모든 경로만큼 p 를 증가시켜주고 단계 3으로 간다.

단계 3 (모형의 해를 구함)

p 개의 경로로 (k-MVASP-p)모형을 세우고 이를 푼다. 단계 4로 간다.

단계 4 (치명호의 최적여부 판정)

만일 (k-MVASP-p)모형의 최적해의 목적함수의 값이 1이면 p = p + 1로 둔다. 단계 2로 간다. 아니면 단계 5 로 간다.

단계 5 (다수치명호 결정)

만일 p 가 1일 경우 다수치명호는 존재하지 않는다. 끝.

p-1 개의 경로로 (k-MVASP)모형을 세우고 이를 푼다.

x_i가 1인 k개의 호들이 다수 치명호가 된다. 끝.

5. 예 제

다음 [그림 1]과 같은 최단경로문제에서 2개의 치명호를 결정하는 문제를 고려해 보자.

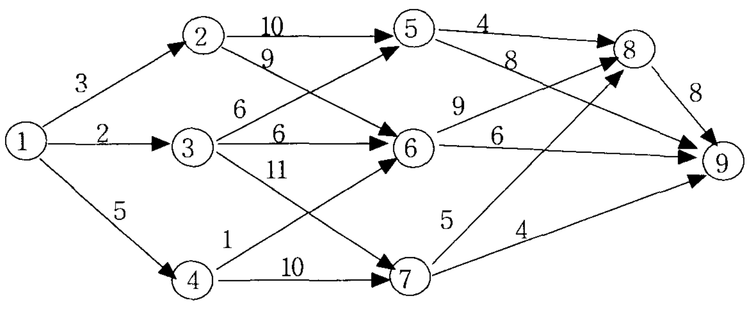
먼저 [그림 1]의 최단경로 문제에서 마디 1→9까지의 모든 경로를 길이의 오름차순으로 나열하면 다음과 같다.

〈표 1〉 경로 및 경로의 길이

변수	경로	경로의 길이(d_j)
r_1	(1 - 4 - 6 - 9)	12.00
r_2	(1 - 3 - 6 - 9)	14.00
r_3	(1 - 3 - 5 - 9)	16.00
r_4	(1 - 3 - 7 - 9)	17.00
r_5	(1 - 2 - 6 - 9)	18.00
r_6	(1 - 4 - 7 - 9)	19.00
r_7	(1 - 3 - 5 - 8 - 9)	20.00
r_8	(1 - 2 - 5 - 9)	21.00
r_9	(1 - 4 - 6 - 8 - 9)	23.00
r_{10}	(1 - 2 - 5 - 8 - 9)	25.00
r_{11}	(1 - 3 - 6 - 8 - 9)	25.00
r_{12}	(1 - 3 - 7 - 8 - 9)	26.00
r_{13}	(1 - 4 - 7 - 8 - 9)	28.00
r_{14}	(1 - 2 - 6 - 8 - 9)	29.00

여기에서 2-MVA는 최단경로로 부터 순차적으로 주어진 경로를 가장 많이 절단하는 호집합을 찾는 문제라고 볼 수 있다.

2-MVA를 구하기 위해 사용된 변수는 <표 2>와 같다.

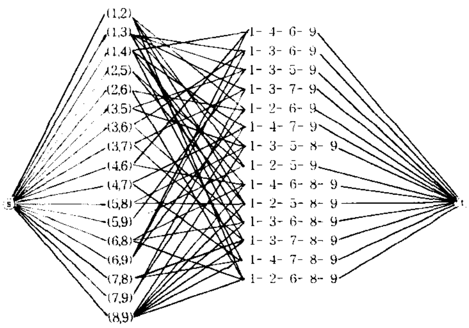


[그림 1] 최단경로문제 예제

〈표 2〉 모형의 예시에서 사용된 변수

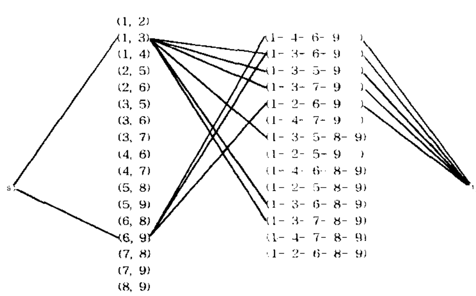
최단경로문제 네트워크 $G=(N,A)$	(k-MVASP)	[그림 2, 3]의 네트워크에서의 의미
호 $(a,b)\in A$	x_i	s에서 출발하는 호
j번째 경로 r_i	z_j	t로 도착하는 호
호와 경로의 연결관계	y_{ij}	그림의 중앙에 나타난 호

이를 통해 [그림 1]의 최단경로문제를 (k-MVASP) 모형으로 나타낸 그림은 [그림 2]이고 최적해는 [그림 3]과 같다.



[그림 2] 모형의 예시

위의 문제의 2개의 치명호는 (1,3), (6,9)이 된다. 즉, (1,3), (6,9)을 동시에 제거할 경우에는 첫 번째 최단경로로부터 다섯 번째 최단경로까지가 절단되어, 2개의 호를 제거하므로 써 최단경로를 6번째 최단경로까지 늘여 줄 수 있는 호가된다.



[그림 3] 모형의 최종 해의 형태

이 결과를 통해 호집합 {(1,3), (6,9)}가 5번째의 최단경로까지를 순서대로 절단하는 호집합으로 최단경로의 길이를 12에서 6번째의 경로의 길이인 19로 늘여 줄 수 있는 2개의 치명호집합이라는 것을 알 수 있다. 이때 [성질 3]과 식 (a)와 식 (c′)에 의해 호집합 {(1,3), (6,9)}가 경로 r_7 (1- 3- 5- 8- 9), 경로 r_{11} (1- 3- 6- 8- 9), 경로 r_{12} (1- 3- 7- 8- 9)에도 속해 있지만 그 이전의 경로인 r_6 (1- 4- 7- 9)가 호집합의 구성요소를 포함하고 있지 않기 때문이다.

6. 결 론

본 연구는 Corley와 Sha의 k-MVA가 가지는 충분조건과 Malik 외 2인이 제시한 다수번째의 최단경로의 해법을 이용하여 기존의 방법으로 최적해를 제공하지 못하는 최단경로문제에서의 k개의 치명호를 찾는 문제의 수리모형 및 알고리즘을 개발 그리고 해법을 효율화 할 수 있는 방법을 제안하는 것이 목적이다.

이를 위해 본 연구에서는 다수개의 치명호를 찾는 문제를 호와 경로와의 관계를 분석하여 구하는 방법을 제시하였다. 먼저 k-MVA는 k개의 치명호가 제거된 상태에서 구한 최단경로의 길이를 이용하여 이 길이보다 짧은 경로들에는 적어도 하나 이상의 치명호가 포함되지만 그렇지 않는 경우의 경로에는 치명호가 포함되지 않음을 보였다. 또한 해법의 효율성을 향상시키기 위하여 시발점과 종착점간의 다수번째 최단경로를 모두 구하지 않고서도 필요할 때 한 경로씩 추가시키면서 계산할 수 있는 방법을 제시하였다.

참 고 문 헌

[1] 박순달, 「OR(경영과학)」, 개정판, 민영사, 1991.
[2] Ahuja R. K., T. L. Magnanti, J. B. Orlin, *Network Flows: Theory, Algorithms, and*

Applications, Prentice-Hall, 1993.

[3] Murty K. G., *Network Programming,* Prentice-Hall, 1992.

[4] Bar-Noy A., S. Khuller, B. Schieber, "The Complexity of Finding Most Vital Arcs and Nodes", *Univ. of Maryland Technical Reports*, CS-TR-3539(1995), pp.1-7.

[5] Corley JR. H. W., D. Y. Sha, "Most Vital Links and Nodes in weighted Networks", *O. R. Letters,* vol.1, No.4(1982), pp.157-160.

[6] Lubore S. H., H. D. Ratliff, G. T. Sicilia, "Determining The Most Vital Link in a Flow Network", *N. R. L. Q.,* Vol.18, No.4 (1971), pp.497-502.

[7] Malik K., A. K. Mittal, S. K. Gupta, "The *k* Most Vital Arcs in the Shortest Path Problem", *Operations Research Letters*, Vol.8(1989), pp.223-227.