韓國國防經營分析學會誌
第 26 卷, 第 2 號, 2000. 12. 31.

수정된 최속경로 알고리즘
(The revised quickest path algorithm)

이 상 욱*, 박 찬 규**, 박 순 달*

Abstract

A quickest path in a network is a path that takes the shortest time to send the amount of data from the source node to the sink node. Martin and Santos presented a theorem on the quickest path by which a quickest path for the amount of data is determined. However, we find a counterexample to Martins and Santos' theorem. In this paper, we present the corrected theorem and give a revised algorithm for finding quickest paths.

* 서울대학교 산업공학과
** 한국전산원 정보화 평가분석단

1. 서　　론

최속경로(quickest path)문제는 Moore [4]에 의해 제안되었으며, Chen과 Chin [3]이 두 개의 목적함수를 갖는 최단경로문제의 변형을 제시하고 나서 본격적인 연구가 시작된 분야이다. 최속경로(quickest path) 문제에서는 각 호의 통과시간과 각 호의 흐름의 용량 상한이 존재하여 주어진 양의 데이터를 시점에서 종점으로 보내는 최소 시간을 갖는 경로를 찾는다. 최속경로(quickest path)문제의 응용분야를 보면, 화물과 운송업체에서 발생하는 물류 문제에서부터 최근 부각되고 있는 통신네트워크 분야에까지 그 범위는 상당히 다양하고 넓다고 볼 수 있다. 최속 경로 문제는 최단 경로 문제와 최대 용량 경로(maximum capacity path)문제의 혼합된 형태이므로 2개의 목적함수를 가진 다목적 최단경로 문제의 일종으로 볼 수 있다. 그러나, 이러한 2개의 목적함수를 동시에 최적화하는 경로는 일반적으로 존재하지 않으므로 하위경로(dominated path)와 우위경로(non dominated path)의 개념을 새로이 도입하게 된다. 우위경로(nondominated path)란 경로상의 지체시간과 용량에 있어서 다른 경로보다 더 우수한 경로를 말하는데, 다목적 계획법의 유효해와 같은 의미를 갖는다. 이러한 우위 경로들 중에서 주어진 데이터의 양을 전송하는데 최소시간이 소요되는 경로를 구하면 이 경로가 최속경로가 된다.

최속경로(quickest path)를 구할 때 경로의 길이와 경로의 최대 유통량을 동시에 고려해야 한다. 최단경로라 할지라도 경로의 용량이 작으면 데이터를 모두 보내는데 소요되는 시간이 다른 경로보다 커질 수 있기 때문이다. 반대로 용량이 최대인 경로일지라도 경로의 길이가 매우 크면 데이터 전송 시간이 다른 경로보다 더 커질 수 있다. 따라서, 어느 한가지만 치중하다 보면 최속경로를 구하지 못할 경우가 발생하게 된다. 즉, 최단시간의 호만 찾다보면, 호의 용량을 고려하지 못하게 되므로 오히려 시간이 더 지체되는 경우가 발생하게되고, 반대로 호의 용량만을 고려하여 큰 용량의 호로 이루어진 경로를 구하면 호의 지체시간을 고려하지 않게 되므로 오히려 경로의 지체시간이 길어질 수 있다. 따라서, 위 두가지의 목적함수를 모두 고려하여야 하며, 이러한 방법으로 경로의 지체시간과 용량에 있어서 다른 경로보다 뛰어난 경로 즉, 우위 경로의 개념을 이용하여 최속경로를 구하게 된다.

본 연구에서는 Martins와 Santos [2]가 제안한 정리에 대한 반례를 제시하고, 수정된 정리와 알고리즘을 제안한다.

2. 최속경로의 수리모형 및 수정된 정리

Martin과 Santos [2]가 제안한 최속경로문제를 정리하면 다음과 같다.

주어진 조건 : 시점 s, 종점 t, 호 (i,j)의 이동시간 t_{ij}, 호 (i,j)의 용량상한 u_{ij}, 전송하려고 하는 데이터의 양 σ

목적 : 시점과 종점사이에 보내려고 하는 데이터의 양 σ를 가장 빠른 시간에 보내는 경로를 찾는 것.

몇 가지 기호를 정의하자. 경로 P를 s와 t사이에 가능한 모든 경로의 집합이라 하고, p를 집합 P에 속하는 하나의 경로, $t(p)$를 경로 p를 지날 때 걸리는 시간, $u(p)$를 경로 p의 용량을 나타낸

다고 하자. 이때, 최속경로의 목적함수는 다음과 같이 나타낼 수 있다.

$$\min_{p \in P} \{ t(p) + \frac{\sigma}{u(p)} \}$$

원래 위의 문제는 다음과 같은 다목적함수로 나누어 생각할 수 있다.

$$\min_{p \in P} t(p) \text{ 과 } \max_{p \in P} u(p)$$

첫 번째 목적함수는 최단경로문제이고 두 번째 문제는 최대용량경로문제로 볼 수 있다. 일반적으로 위의 두 목적함수를 동시에 만족시키는 최적해는 존재하지 않는다. 따라서 하위 경로(dominated path)와 우위 경로(nondominated path)라는 개념을 사용하게 되는데, 우위 경로란 두 개의 경로 $p, q \in P$가 주어졌을 때, $t(p) \le t(q)$ 이고 $u(p) > u(q)$ 이거나, 또는 $t(p) < t(q)$ 이고 $u(p) \ge u(q)$ 일 때의 경로 p를 말하며, 경로 p가 경로 q를 우위한다(dominate)고 한다. 기호로는 pDq를 사용한다. 하위경로란, P를 $P_D = \{p \in P | \exists q \in P, qDp\}$ 와 $P_N = P - P_D$로 나누었을때, P_D에 속하는 경로를 말한다. P_N에 속하는 경로는 우위 경로가 된다. 결국, 주어진 데이터의 양에 대한 최속경로란 지체시간은 작고 용량은 최대인 경로를 찾는 것이므로, 최속경로는 이러한 우위 경로 집합 중에서 존재하게 되며, 특정 데이터의 양에 대한 최속경로는 우위경로가 됨을 Martins와 Santos [2]가 증명하였다.

우위 경로집합 P_N을 $i \in \{1, \cdots, r-1\}$에 대하여 $t(p_i) \le t(p_{i+1})$ 이고 $u(p_i) \le u(p_{i+1})$인 규칙으로 정렬을 하자. 정렬된 결과를 $P_N = \{p_1, \cdots, p_r\}$ 이라고 하자. 위와 같이 하면, 집합자체가 우위 경로 집합이기 때문에 $u(p_i) = u(p_{i+1})$ 인 경우에만 $t(p_i) = t(p_{i+1})$ 이 된다. 이러한 성질을 이용하여 Martins와 Santos [2]는 다음의 정리를 내놓았다.

(정리)

(1) $\sigma \in]0, \frac{t(p_2) - t(p_1)}{u(p_2) - u(p_1)} \times u(p_1) \times u(p_2)]$ 에 대하여 p_1은 최속경로이다.

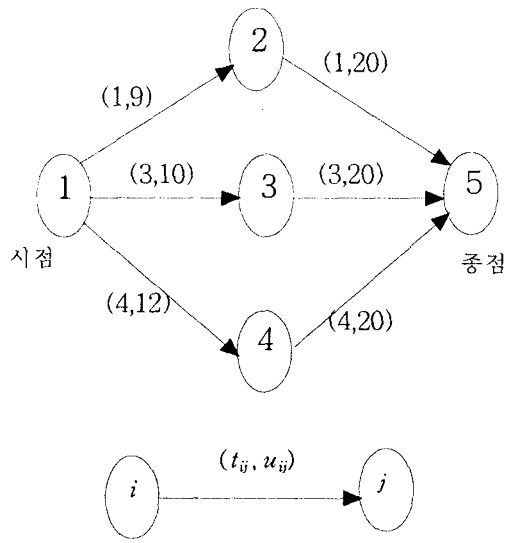
(2) $\sigma \in [\frac{t(p_i) - t(p_{i-1})}{u(p_i) - u(p_{i-1})} \times u(p_{i-1}) \times u(p_i), \frac{t(p_{i+1}) - t(p_i)}{u(p_{i+1}) - u(p_i)} \times u(p_i) \times u(p_{i+1})]$

$i \in \{2, \cdots, r-1\}$

에 대하여 p_i는 최속경로이다.

(3) $\sigma \ge \frac{t(p_r) - t(p_{r-1})}{u(p_r) - u(p_{r-1})} \times u(p_{r-1}) \times u(p_r)$ 에 대하여 p_r은 최속경로이다.

위의 정리에 위배되는 다음의 (그림 1)의 예가 존재한다.



(그림 1) 예

(그림1)을 보면, 시점에서 종점까지의 우위 경로 3개가 존재한다.

경로	지체시간	용량
$p_1(1\to2\to5)$	2	9
$p_2(1\to3\to5)$	6	10
$p_3(1\to4\to5)$	8	12

위의 정리에 의하면 경로 p_1은 $\sigma\in]0,360]$에 있어서 최속경로이어야 한다. 그러나, $\sigma=300$인 경우를 보면,

$$t(p_1)+\frac{\sigma}{u(p_1)}=35+\frac{1}{3}>33=t(p_3)+\frac{\sigma}{u(p_3)}$$

이 성립된다. 즉, 경로 p_3가 최속경로가 된다. 따라서, 위의 정리는 다음과 같이 수정되어야 한다.

(수정된 정리)

$\alpha(x,y)$를 다음과 같이 정의하자.

$$\alpha(x,y)\ :\ \frac{t(p_x)-t(p_y)}{u(p_x)-u(p_y)}\times u(p_x)\times u(p_y)$$

그러면, 위의 정리는 다음과 같이 변경된다.

(1) $\sigma\in]0,\ \min_{2\le k\le r}\alpha(k,1)]$에 대하여 p_1은 최속경로이다.

(2) $i\in\{2,\cdots,r-1\}$에 대하여

$\max_{1\le k\le i-1}\alpha(i,k)\le\min_{i+1\le j\le r}\alpha(j,i)$이면,

$\sigma\in[\max_{1\le k\le i-1}\alpha(i,k),\ \min_{i+1\le j\le r}\alpha(j,i)]$에 대하여 경로 p_i는 최속경로이다.

$\max_{1\le k\le i-1}\alpha(i,k)>\min_{i+1\le j\le r}\alpha(j,i)$이면, 임의의 $\sigma\in R^+$에 대하여 경로 p_i는 최속경로가 아니다.

(3) $\sigma\ge\max_{1\le k\le r-1}\alpha(r,k)$에 대하여 경로 p_r은 최속경로이다.

(증명)

경로 p_1이 최속경로가 되기 위해서는, 정의에 의해서

$$t(p_1)+\frac{\sigma}{u(p_1)}\le t(p_k)+\frac{\sigma}{u(p_k)},\ k=2,\cdots,r \quad (1)$$

이 만족되어야 한다. 위의 식은 모든 $j\in\{1,\cdots,r-1\}$에 대하여 $t(p_j)<t(p_{j+1})$이고, $u(p_j)<u(p_{j+1})$이기 때문에 다음과 같이 쓰여질 수 있다.

$$\sigma\le\frac{t(p_k)-t(p_1)}{u(p_k)-u(p_1)}\times u(p_k)\times u(p_1),\ k=2,\cdots,r \quad (2)$$

수정된 정리의 (1)번은 위의 식으로부터 유도될 수 있다.

경로 $p_i(1<i<r)$가 최속경로가 되기 위해서는, 다음의 부등식이 성립되어야 한다.

$$t(p_i)+\frac{\sigma}{u(p_i)}\le t(p_k)+\frac{\sigma}{u(p_k)},\quad k=1,\cdots,i-1,i+1,\cdots,r. \quad (3)$$

(2)번에서와 마찬가지로 위의 식은 다음과 같이 나타낼 수 있다.

$$\sigma\ge\frac{t(p_i)-t(p_k)}{u(p_i)-u(p_k)}\times u(p_i)\times u(p_k),\ k=1,\cdots,i-1,$$
$$\sigma\le\frac{t(p_k)-t(p_i)}{u(p_k)-u(p_i)}\times u(p_k)\times u(p_i),\ k=i+1,\cdots,r \quad (4)$$

위의 식은 동시에 만족되어야 하므로 (수정된 정리)의 (2)번은 쉽게 유도되어질 수 있다.

경로 p_r이 최속경로가 되기 위해서는 다음의 부등식이 만족되어야 한다.

$$t(p_r)+\frac{\sigma}{u(p_r)}\le t(p_k)+\frac{\sigma}{u(p_k)},\ k=1,\cdots,r-1 \quad (5)$$

그리고 (5)는 다음의 식으로 나타낼 수 있다.

$$\sigma\ge\frac{t(p_r)-t(p_k)}{u(p_r)-u(p_k)}\times u(p_r)\times u(p_k),\ k=1,\cdots,r-1 \quad (6)$$

수정된 정리의 (3)번 항은 위의 식으로부터 유도

되어질 수 있다.

수정된 정리를 (그림 1)에 적용하면, $\sigma \in]0, 216]$ 에 대하여는 p_1이 최속경로이고, $\sigma \geq 216$인 경우에는 p_3이 최속경로가 된다. 경로 p_2는 어떤 $\sigma \in R^+$ 에 대하여도 최속경로가 되지 못한다.

3. 알고리즘

시점과 종점이 주어지고 전송할 데이터의 양이 주어진 경우에서의 최속경로를 구하는 알고리즘은 다음과 같다. 이 방법은 시점과 종점간의 모든 우위경로로 이루어진 집합 P_N을 구하지 않고, 일부의 우위경로들 중에서 최속경로를 찾는 방법이다. 우위경로를 찾기 위해 최단경로 p_1을 구하고 이 경로의 용량 $u(p_1)$보다 작거나 같은 호는 제거한다. 남아있는 호의 용량은 모두 $u(p_1)$보다 크게 된다. 다시 시점과 종점간의 최단경로 p_2를 구한 후, $u(p_2)$보다 작거나 같은 호들은 제거한다. 같은 방식으로 더 이상 경로가 존재하지 않을 때까지 반복하면 순차적으로 구하여진 경로 $p_1, p_2, \cdots, p_r$은 정렬된 우위경로들이 된다. 정렬된 우위경로들은 '최속경로는 우위경로'이라는 것과 '수정된 정리'에 의하여 구간별로 나누어지는 데이터의 양들에 대한 최속경로가 된다. Martins와 Santos [2]는 이러한 우위경로들을 찾기 위해 꼬리표법에 기반을 둔 방법을 사용하였다. 마디 i에 대한 꼬리표는 $(\pi_i, \gamma_i, \chi_i)$를 사용하는데, π_i는 시점 s로부터 마디 i까지의 경로에 대한 지체시간의 상한이며, γ_i는 s로부터 i까지의 경로에 대한 용량의 하한, χ_i는 마디 i의 선행 마디를 의미한다. 단계4에서 '수정된 정리'를 사용하여 주어진 데이터에 대한 최속경로를 구하게 된다.

단계1. 초기화

σ: 전송할 데이터의 양. $\pi_i = \infty$, $\gamma_i = \infty$, $\chi_i = 0$, $i \in N$, s: 시점, t: 종점, $\pi_s = 0$, $\gamma_s = 0$, $k = 1$, $i = s$

단계2. $s-t$ 최단경로 선정

꼬리표법을 사용하여 시점과 종점간의 최단경로를 구한다. 마디에 대한 꼬리표의 수정은 다음과 같다. 마디 i의 이웃 마디 j에 대하여, $\pi_i + t_{ij} < \pi_j$ 이면, 마디 j의 꼬리표를 $(\pi_i + t_{ij}, min\{\gamma_i, u_{ij}\}, i)$로 정한다. $\pi_i + t_{ij} = \pi_j$이고 $min\{\gamma_i, u_{ij}\} > \gamma_j$이면, 마디 j의 꼬리표를 $(\pi_j, min\{\gamma_i, u_{ij}\}, i)$로 정한다. 단계3으로 간다.

단계3. 호의 제거

시점과 종점간에 더 이상 경로가 없으면 단계4로 간다. 아니면, $G(N, A)$에서 $u_{ij} \leq u(p)$인 모든 호를 제거한다. $k = k+1$로 하고 단계2로 간다.

단계4. 경로 선정

'수정된 정리'를 사용하여, 구해진 각각의 우위경로들에 대하여 최속경로가 되는 데이터 양의 범위를 산정하고, 주어진 데이터에 대한 최속경로를 선정한다.

종료

알고리즘의 복잡도(complexity)는 네트워크의 호의 수를 m, 마디수를 n으로 할 때, 최단경로는 많아야 m번 풀게 되고, 최단경로는 $O(nlog(n)+m)$에 풀 수 있으므로, 복잡도는 $O(mnlog(n)+m^2)$

[2]이 된다.

4. 예　　제

위의 (그림1)을 예제로 최속경로를 구해보자.

<1회>

단계1. 초기화

$\sigma = 300$, $\pi_i = \infty$, $\gamma_i = \infty$, $\chi_i = 0$, $i \in N$

$\pi_1 = 0$, $\gamma_1 = 0$, $k = 1$, $i = 1$

단계2. $s - t$ 최단경로 선정

마디 1의 이웃마디인 2, 3, 4에 대하여 각각의 꼬리표를 $(1, 9, 1)$, $(3, 10, 1)$, $(4, 12, 1)$로 수정한다. 마디 2가 최소 시간을 가지므로 마디 2의 이웃마디 5에 대하여 꼬리표를 수정한다. $(2, 9, 2)$가 된다.

마디 5는 종점마디이므로 단계 3으로 간다. $p_1 : 1-2-5$, $u(p_1) = 9$, $t(p_1) = 2$

단계3. 호의 제거

단계2에서 구해진 경로 p_1의 용량보다 작거나 같은 호들을 제거한다. 호 $(1,2)$가 제거된다. 시점과 종점간에 경로가 존재하므로 $k = 2$로 두고, 단계 2로 간다.

<2회>

단계2. $s - t$ 최단경로 선정

마디 1의 이웃마디 중 마디 3이 최소시간을 가지므로 마디 3의 이웃마디 5의 꼬리표를 $(6, 10, 3)$로 바꾼다. 마디 5가 종점마디이므로 단계 3으로 간다.

$p_2 : 1-3-5$, $u(p_2) = 10$, $t(p_2) = 6$

단계3. 호의 제거

단계2에서 구해진 경로 p_2의 용량보다 작거나 같은 호들을 제거한다. 호 $(1,3)$이 제거된다. 시점과 종점간에 경로가 존재하므로 $k = 3$으로 두고 단계 2로 간다.

<3회>

단계2. $s - t$ 최단경로 선정

마디 1의 이웃마디 중 마디 4가 최소시간을 가지므로 마디 4의 이웃마디 5의 꼬리표를 $(8, 12, 4)$로 바꾼다. 마디 5가 종점마디이므로 단계 3으로 간다.

$p_3 : 1-4-5$, $u(p_3) = 12$, $t(p_3) = 8$

단계3. 호의 제거

단계2에서 구해진 경로 p_3의 용량보다 작거나 같은 호들을 제거한다. 호 $(1,4)$가 제거된다. 시점과 종점간에 경로가 존재하지 않으므로 단계 4로 간다.

단계4. 경로 선정

경로 p_1은 $\sigma \in]0, 216]$에 대하여 최속경로.

경로 p_2는 어떤 데이터에 대하여도 최속경로가 되지 않는다.

경로 p_3는 $\sigma \geq 216$에 대하여 최속경로이다. 주어진 데이터의 양이 300이므로 p_3가 최속경로가 된다. 종료

5. 결　　론

본 연구에서는 최속경로의 정의와 우위 경로 및 이에 따른 정리들을 살펴보았다. 또한, Martins과 Santos [2]가 제안한 정리에 대한 반례를 제시하였고 이에 대한 수정된 정리와 알고리즘을 제시하였다. 요즘은 LAN 및 인터넷의 확장 등으로 네트워크의 복잡도가 증가하고 이에 따라 데이터의 전송에 따른 빠른 시간의 경로를 구하는 것이 중요한 과제

로 부각되고 있다. 따라서, 이러한 네트워크에서의 데이터의 전송에 최속경로 알고리즘을 적용하면 좀 더 효율적인 네트워크의 활용이 가능할 것으로 기대된다.

참고문헌

[1] 박순달, 경영과학, 제3판, 민영사, 1998

[2] Ernesto de Queiros Vieira Martins, Jose Luis esteves dos Santos. "An algorithm for the quickest path problem", *Operations Research Letters* 20(1997), 195-198.

[3] Y. L. Chen and Y.H. Chin. "The quickest path problem", *Computers & Operations Research* 17(1990), 153-161.

[4] M. H. Moore, "On the fastest route for convoy-type traffic in flowrate-constrained networks", *Transportation Science* 10(1976), 113-124.