General networks 에 있어서 최단 경로 문제에 대한 연구
- A Study on the Shortest Path Problem in General Networks -

김 준 홍*
Kim, Jun Hong

Abstract

Finding shortest paths in networks is the fundamental problem in network theory and has numerous in Operations Research and related fields. The purpose of this study is to present a algorithm for solving the length of the shortest paths from a fixed node in a general network in which the arc distance can be arbitrary value. This algorithm has a worst computational bound of $n^3/4$ additions and $n^3/4$ comparisons, which is lower the the worst computational bounds of other available algorithms.

1. 서론

network 에서 최단 경로를 구하는 문제는 수송 및 정보 통신 network 분야에서도 가장 기본적이고 자주 부닥치는 문제이며, O.R. 과 그 관련 분야에서 많이 응용되고 있다. 최단 경로 문제에 대해 효율적인 앨고리즘의 설계 및 검정은 network 최적화 분야에서 중요하다. 최단 경로 문제의 대상 모델로는 여러 가지가 있지만[1][6], 이 논문에서는 두 지점간의 길이가 음(陰)이 존재하는 일반적인 network을 대상으로 한다. 이 모델에서 한 정점에서 다른 모든 정점으로의 최단 경로를 구하는 문제는 상대적으로 비효율적이고 때로는 복잡한 앨고리즘으로 만이 해결될 수 있다.
이 문제에 대한 앨고리즘은 Ford[10]가 최초로 제시한 이래, Danzig,G.L.[3], Bellman[2], Floyd[9], Moore[13] 에서는 앨고리즘의 이론적인 성질들의 연구와 새로운 앨고리즘의 개발에 헌신하였으며, Edmonds[5]는 이러한 형태의 앨고리즘은 결국 nonpolynomial time 이라 보였다. 그 후, Kershenbaum[12], Glover, Klingman, Phillips, and Schneider[11] 는 이 문제에 대한 O(nm) time 을 제시하였고, 여러 연구에서 polynomial time primal simplex 앨고리즘에 관심을 두었다[1].
이 논문에서는 계산상 최선의 앨고리즘 제시 보다는 general network 이 갖는 복잡성에 비해 상대적으로 간단할 절차에 의해 해에 이르는 앨고리즘의 제시에 주안점을 두어 이 앨고리즘의 수렴성을 증명하였고, 앨고리즘의 효율성을 검정하기 위해 Danzig, Blattner, and Rao[5], Bellman[2], Ford[11], Moore[14], 그리고 Floyd[10] 와 Danzig[4]의 앨고리즘과 비교하였다. 특히, Danzig[4] 앨고리즘과는 100개 정도의 문제에 대해 그 연산 시간을 비교 검정하였다.

2 General network 에 대한 제안 앨고리즘

2-1 부호와 가정

Directed network $N=(V,E,c)$를 생각한다: 여기서, $V=\{1,2,..,n\}$은 정점들의 집합, E는 선분$(i,j),\forall i,j$,들의 집합, $c_{ij}\in R\cup\{\infty\}$ $\forall(i,j)\in E$, $|V|=n$, $V\cap E=\emptyset$ 이다. 길이 행렬 D(N) 의 원소 d_{ij} 를 다음과 같이 정의한다:

* 수원대학교 산업공학과

$$d_{ij} = \begin{cases} 0, & i=j \text{ 인 경우} \\ d[i,j], & i \text{ 와 } j \text{ 가 서로 직접 연결되어 있는 경우,} \\ \infty, & \text{이외의 경우} \end{cases}$$

network에서(topological ordered 되어 있다 하자) 시작 정점 s=1 에서 끝정점 t=n 까지의 경로는 순서열 (1, 2, ..., n) 으로 나타낸다. 여기서 s=1, t=n, $i \neq i+1$, for i=1, 2, ..., n-1.

경로 s=1에서 t=n까지의 길이는 $f_n = d_{12} + d_{23} + \ldots + d_{n-1,n}$ 로 나타내고, f_i, $i \in N$, 는 1에서 정점 i 까지의 최단 경로의 길이, $f_i(k)$ 는 앨고리즘의 k 번째 반복 단계에서 정점 1 에서 i 까지의 임시 최단 경로의 길이라 한다. 경로의 길이가 음(陰)이고, s=t 라면, 그 경로는 negative cycle이라 한다.

$f_i(k)$ 는 초기화되어 있으며, 그 값이 최종적으로 확정되지 않았을 때는 반복 단계 k에서 변경된다. 최종 경로는 $\bar{f}_i(k)$ 로 나타낸다.

$f = (f_1, f_2, \ldots, f_n)$,
$f(k) = (f_1(k), f_2(k), \ldots, f_n(k))$,
$\bar{f}(k) = (\bar{f}_1(k), \bar{f}_2(k), \ldots, \bar{f}_n(k))$ 라하고,

$z(k)$는 모든 i 에 대해 $f_i(k) = \bar{f}(k)$ 일 때의 the number of coordinates 라 한다. 여기서 $\bar{f}(k)$와 $\bar{f}(k-2)$는 반복 단계 k 에서 서로 일치한다.

2.2 제안 앨고리즘

general network에서 임의 정점에서 다른 모든 정점까지의 최단 경로를 구하기 위한 앨고리즘을 제시한다. 이 앨고리즘의 장점은 많아야 $n^3/4$ additions 과 $n^3/4$ comparisons 이 요구된다. 이것은 다른 앨고리즘의 worst computational bounds 보다 더 적을 값이다. 정점 1 에서 모든 다른 정점으로의 최단 경로의 길이를 구하기 위해, 또는 negative cycle의 존재성 여부를 추적하기 위해 제안 앨고리즘은 다음과 같이 진행한다:

<u>앨고리즘</u>

```
given distance matrix D(N)
let  f(-1)=(∞,∞,...,∞) and f(0)=(0,∞,∞,...,∞)
iteration  k=1,3,5,...
  let z(k)=0 , and f(k)=f(k-1)
  for j=1, 2, ..., n-1 do;
   for i=j+1,j+2,..., n do;
     if fj(k)<fj(k-2), fi(k):= min ( fi(k), fj(k)+dji )  such that dji≠∞.
  let q=k-1
  if fi(q)=fi(q-1), z(k)=z(k)+1
iteration k=2,4,6,...
  let z(k)=0 ,and f(k)=f(k-1)
  for j=n,n-1,...,2 do ;
   for i=1, 2, ..., j-1 do ;
     if fj^(k)<fj^(k-2), fi^(k)= min (fi^(k), fj^(k)+dji ) such that dji≠∞.
  let q=k-1
  if  fi(q)=fi(q-1), z(k)=z(k)+1
stopping rule
  stop after iteration k if  (1) z(k)=n, (2) z(k)<k-1, or (3) f1(k)<0.
```

참고: stoppig rule (1) 은 f(k)=f 라는 것을 의미하고, stopping rules (2) 와 (3) 은 negative cycle이 존재함을 나타낸다.

다음 장에서 증명하는 바와 같이 negative cycle이 존재하지 않는다면, 매 반복 단계에 대해 $z(k) \geq k-1$ 이다. 따라서 stopping rule (2) 의 관점에서 볼 때, 제안 앨고리즘은 최대한 n+1 반복 단계를 요구한다. 그 다음 장에서는 제안 앨고리즘은 많아야 근사적으로 $n^3/4$ additions 과 $n^3/4$ comparisons 을 필요로 함을 보인다.

2.3 제안 앨고리즘의 수렴성

이 앨고리즘의 수렴성을 증명하기 위해 network의 monotone order를 정의하겠다.
$(p_1, p_2, \ldots, p_{s_m})$은 정점 1에서 $r \neq 1$ 까지의 경로라 하자. 다음과 같이 path를 그룹화 하자;

$$
\begin{aligned}
1 = p_1 &< p_2 < \ldots < p_{s_1}, \\
p_{s_1} &> p_{s_1+1} > \ldots > p_{s_2}, \\
&\ldots, \\
p_{s_m-1} &< \ldots < p_{s_m} = r
\end{aligned}
$$

경로의 monotone order $M((p_i))$를 이들 group들의 수라 하자. 즉, $M((p_i))=m$, 예를 들면 $(p_1,p_2,\ldots,p_6)=(1, 5, 7, 2, 4, 3,)$ 이라면, $M((p_i))=4$.
지금 network에 negative cycle이 없다 하자. $f_r < \infty$ 인 정점 $r(\neq 1)$ 에 대해, r의 monotone order $M((r)) := \min_{(p_i)} [M((p_i))]$ 라 정의한다. 여기서 (p_i)는 r 까지의 최단 경로로 언급된다.

$r=1$ 또는 $f_r = \infty$ 라면, $M(r)=0$ 이라 한다.

또, network의 monotone order $M^* := \max_r [M(r)]$ 라 정의하자. 여기서 r은 network의 nodes 들이 된다.

그러면, 다음 lemma 를 제시할 수 있다.

<u>Lemma 1</u>
network 의 각 node $i= 0,1,\ldots,M^*$에 대해 *monotone order* M^* 를 갖는다면, $M(r)=i$ 를 만족하는 node r이 존재한다.

다음 정리는 반복 단계 $M(j)$ 에서 정점 j 가 최종적 labeling 된다는 것을 제시한다.

<u>정리 2</u>
network가 monotone order M^* 을 갖고 있고, $k=0,1,\ldots,M^*$ 와 r 이 임의의 정점이라면

(i) $M(r) \leq k$ 라면, $\bar{f}_r(k)=f_r$

(ii) $M(r) > k$ 라면, $\bar{f}_r(k) > f_r$ 이다.

(증명)
k에 대해 induction으로 진행한다.: $k=0$에 대해 이 결과는 trivial.
$k \leq h$에 대해 이것이 사실이라면 $k=h+1$에 대해 이 정리를 증명하자.
h가 홀수이거나 짝수가 되는 두 가지 경우가 있지만 둘다 동일한 결과를 제시한다.
h가 짝수라 하자.
(i) r은 $M(r)=h+1$인 정점이라 하자. monotone order $M(r)$의 r에 대해 최단 경로 (p_i)가 존재한다. 전과 같은 grouping을 갖는다.

$$
\begin{aligned}
1 = p_1 &< p_2 < \ldots < p_{s_1}, \\
p_{s_1} &> p_{s_1+1} > \ldots > p_{s_2}, \\
&\ldots \\
p_{s_{m-1}} &< p_{s_{m-1}+1} < \ldots < p_{s_m} = r.
\end{aligned}
$$

$(q_0, \ldots, q_t) = (p_{s_{m-1}}, \ldots, p_{s_m})$이고 $t=s_m-s_{m-1}$이라 하자. q_0 는 monotone order h 라하고 $q_i, i=1,2,\ldots,t$ 는

$f_{q_1}=f_{q_0}+d_{q_0 q_1}$으로 대치된다. $j=q_1,q_2,\ldots,q_{t-1}$일 때, $f_{q_i}(h+1) < f_{q_i}(h-1)$ 을 갖고 따라서, $f_{q_{i+1}}(h+1)$은 $f_{q_i+1}=f_{q_i}+d_{q_i q_{i+1}}$로 대치된다. 특히, $j=q_{t-1}$일 때, $f_{q_t}(h+1)$은 f_{q_t} 가되고 (i) 은 증명된다.

(ii) $\bar{f}_r(h+1)=f_r$ 인 $M(r)>h+1$을 갖는 정점 r이 있다 하자. 여기서, r 은 가장 작은 그러한 정점이라 하자. $\bar{f}_r(h)=f_r$ 이므로, 반복 단계 h+1 에서 $j<i$ 에 대해 $f_r(h+1)$은 $f_r = \bar{f}_j(h+1)+d_{ij}$ 로 변경된다. 이것은 $\bar{f}_j(h+1) < \bar{f}_i(h-1)$ 과 $\bar{f}_j(h+1)=f_j$ 라는 것을 의미한다. 가정과 함께 $M(j)=h$ 또는 h+1이 된다. 두 경우 중 하나에서 h 는 짝수이므로 $M(r)=h+1$ 을 갖는다. 이것은 우리의 가정에 어긋나고 (ii)는 증명이 된다.

<u>corollary 3</u>

network 가 negative cycle을 갖지 않는다면, $\bar{f}(k)$의 coordinate $\bar{f}_j(k)$ 중 적어도 k+1 개는 최종적인 값을 갖는다. 즉 $\bar{f}_j(k)=f_j$.

(증명)

매 반복 단계 $k=0,1,\ldots,M^*$ 에 대해 적어도 node 한개는 최종적으로 labeling 된다는 것을 정리는 언급하고 있다.

<u>corollary 4</u>

network 가 negative cycle을 갖지 않는다면, k=1, 2, ..., n+1 에 대해 $z(k) \geq k-1$ 이다.

<u>corollary 5</u>

$z(k) < k-1$ 이라면, negative cycle 이 존재한다.

<u>corollary 6</u>

negative cycle이 존재하지 않는다면, 앨고리즘은 정확히 M^*+2 반복 단계 내에 종료된다.

(증명)

반복 단계 M^*+1 과 M^*+2 에서 변동이 생기지 않고, 따라서 $z(M^*+2)=n$ 이다.

3. 제안 앨고리즘의 효율성

앨고리즘을 수행하기 위해 요구된 연산의 근사값의 식으로 제안 앨고리즘의 효율성을 파악한다. 제안 앨고리즘이 모든 j=1,2,...,n 에 대해 $f_j(k) < f_j(k-2)$ 이라면, 반복 회수 k=1,2,... 를 수행하기 위해 각각 $(n-1)^2/2$ additions 와 comparisons 을 요구된다. 그러나, network 에서 negative cycle이 존재하지 않는다면, 적어도 $z(k) \geq k-1$ 인 i 에 대한 f_i 는 적어도 $z(k-2) \geq k-3$ 인 j 에 대해 $f_j(k)=f_j(k-2)$ 라는 조건을 만족하는 반복 단계 k 에서 최종적으로 labeling 이 된다. $f_j=f_j(k-2)$ 하나를 결정하는 것은 반복 단계 k 에서 두개의 반복 단계 각각에서 n-1 개의 additions 과 comparison 을 절약하게 되므로, 반복 단계 k 에서 앨고리즘은 근사적으로 $n^2/2 - n\cdot z(k-2)/2$ 개의 additions 과 comparisons 을 요구한다. 앨고리즘은 최대한 M^*+2 회의 반복 단계에서 중지하므로, 앨고리즘은 근사적으로 $(M^*+2)\cdot n^2/2 - n\cdot[\sum_{k=2}^{m^*+2} z(k-2)]/2$ 개의 addition과 comparison을 요구한다. 이것은 $M^*=n-1$ 일 때 근사적으로 $n^3/4$ additions과 $n^3/4$ comparison 의 상한을 갖는다. 그러나 network에 negative cycle이 있다면 stopping rules (2) 와 (3) 은 반복 단계 n 으로 앨고리즘을 종료시킬 것이다. 따라서, 그러한 경우 앨고리즘은 더 적은 연산 회수를 요구할 수 있다.

4. 앨고리즘의 비교

제안 앨고리즘을 수행하는데 요구된 연산 회수의 식으로 제안 앨고리즘과 다른 앨고리즘들, 어기서는 앨고리즘의 방법과 그 연산 회수가 유사한 그룹으로 나누어, 첫 번째 그룹으로 Danzig, Blattner, and Rao[5], 두 번째 그룹으로 Bellman[2], Ford[10], Moore[13], 세 번째 그룹으로 Floyd[9], Danzig[4], 과의 효율성을 비교해 본 후, 다음은 그 비교표이다.

표1. 효율성 비교 [주1)]

앨고리즘	addition과 comparison의 근사치	addition과 comparison의 상한의 근사치
제안 앨고리즘	$(M^{*}+2)n^{2}/2 - [\sum_{k=3}^{m^{*}+2} z(k-2)]\cdot n/2$	$n^{3}/4$
Danzig,Blattner, and Rao's	$(3\sum_{k=1}^{M^{**}} k^{2})/2$	n
Bellman's , Ford's , and Moore's	$M^{***}n^{2}$	n^{3}
Floyd's and Danzig's	n^{3}	n^{3}

Dreyfus[7]가 제안한 것처럼, Danzig, Blattner, and Rao 의 앨고리즘에서 계산상의 상한은 제안 앨고리즘의 상한보다 약간 더 크지만, 이들 앨고리즘들이 실 문제에 얼마나 빨리 수렴하는 가는 미지수이다. 다음 표2 는 이들 두 앨고리즘들의 효율성을 비교하기 위해 실행한 실험 결과이다. 프로그램 언어는 FORTRAN 을 사용하였다. 첫째, 이 프로그램에서 연산 회수에 대한 계산은 앨고리즘을 수행하는데 필요한 총 연산 회수의 정보를 얻기 위해 수행하였다. 이에 필요한, random number generator는 100 node network의 거리가 되는 random number 들은 일양확률분포를 갖는 δ 와 $\delta+100$ 사이의 정수를 난수로 하여 발생시켰다.

표2 제안 앨고리즘과 Danzig, et al 에 의한 앨고리즘 효율성의 비교

비음거리의 비율	앨고리즘	구분	loops 의 수	additions 의 수	substraction 의 수	comparisons 의 수	transfers 의 수	연산 총수	비율
1%	제안 앨고리즘	평균	18,110	17,730	360	17,660	18,377	72,239	0.5
		표준편차	10,043	10,056	216	10,000	10,487	41,001	
	Danzig's 앨고리즘	평균	37,515	16,666	14,001	51,286	23,865	142,355	
		표준편차	20,779	8,629	7,648	28,398	11,352	76,771	
1.5%	제안 앨고리즘	평균	11,314	11,064	221	11,043	11,394	45,038	0.56
		표준편차	6,299	6,188	117	6,155	6,412	25,173	
	Danzig's 앨고리즘	평균	20,701	9,098	7,571	28,075	13,842	79,287	
		표준편차	11,128	4,520	3,992	15,101	6,197	40,914	
2%	제안 앨고리즘	평균	7,948	7,756	161	7,747	7,983	31,596	0.77
		표준편차	5,921	5,827	105	5,808	5,988	23,651	
	Danzig's 앨고리즘	평균	10,945	4,808	3,912	14,686	7,875	42,228	
		표준편차	7,144	2,849	2,491	9,620	4,203	26,297	

주1 $1<M^{*},M^{**},M^{***}<n-1$ 이고 특정 network 에 좌우된다.
또 특정 network 에 따라, $M^{*} \le M^{***}$, $M^{*}<$ or $>M^{**}$ 이다.

실험에서 두 앨고리즘은 100 node network 에 대해 각각 20 회 적용되었다. 여기서, δ 는 상수이고 정점 간의 거리는 20 배로 랜덤하게 변하게 하였다. 이들 20 배수에는 3 block 을 두었다(물론 각 배수에서 두 앨고리즘을 비교하고 있다). 여기서, δ 은 block에 대해 변하고, δ 는 -10, -15, -20 의 값으로 하였다. 이것은 negative distance 의 비율을 각각 1% 에서 1.5% 와 2% 까지 변하게 한다.

δ 가 -20 보다 더 적을 때, network은 negative cycles로 abundant가 되어서 두 앨고리즘 모두 거의 즉시 중단된다. 따라서 -20 보다 더 적은 δ 에 대해 비교하는 것은 의미가 없다. 다음은 δ 를 -10, -15, -20 으로 취한 것에 대해 20 개의 문제를 수행했을 때 두 앨고리즘으로 부터 얻어진 연산 회수의 평균치와 표준편차이다.

표2 에서 볼 수 있는 바와 같이, 제안 앨고리즘과 Danzig et al 의 앨고리즘은 network 에 있는 길이들이 상당히 많이 negative 일 때 유사한 효율성을 가짐을 알 수 있다. 그것은 그 경우 negative cycle 이 많이 존재하는 이유이다. 그러나, 음의 길이를 갖고 있는 선분이 적을 때 (이러한 경우, 소수의 negative cycle 이 존재하거나 거의 존재하지 않는다) 제안 앨고리즘은 Danzig et al 의 앨고리즘에 비해 그 효율성은 월등함을 보여준다. 그러나, 이 실험은 특정한 형태의 network에 대해서 실행되었음을 생각할때, 이들 결과는 모든 경우, 특이한 형태의 network, 에 다 적용된다고 볼 수 없다.

이 논문은 General network 에서의 최단 경로를 구하기 위한 최선의 앨고리즘을 제시하고자 하는 의도라기 보다는, 이 모델이 갖고 있는 특수성에 비해, 제안 앨고리즘은 비교적 간단한 절차를 통해 최단 경로를 구할 수 있고, 최악의 경우에도 그 연산 회수는 적어도 다른 앨고리즘 보다 더 적을 수 있다.

참 고 문 헌

1. Ahuja,R.K., T.L. Magnanti, and J.B. Orlin, *Network flows*, Prentice-Hall, U.S., 1993.
2. Bellman,R.E., *On a routing problem*, Quart. Appl. Math., Vol.16, No.1, P.87-90, 1958.
3. Danzig,G.B., *On the shortest route through a network*, Man. Sci., Vol.6, No.2, P.187-190, 1960.
4. Danzig,G.B., *All shortest routes in a graph*, (Ed.)P.Rosentiehl, *Theory of Graphs*, P.85-90, Gordon and Breach, New York, 1967.
5. Danzig,G.B., W.O. Blattner, and M.R. Rao, *All shortest routes from a fixed origin in a graph*, (Ed.)P.Rosentiehl, *Theory of Graphs*, P.85-90, Gordon and Breach, New York, 1967.
6. Domschke,W., *Logistik:Transport*, R.Oldenbourg Verlag, Germany, 1981.
7. Dreyfus,S.E., *An appraisal of some shortest path algorithms*, Operations Research, Vol. 17, No. 3, P. 395-412, 1969.
8. Edmonds,J., *Exponential growth of the simplex method for the shortest path problem*, University of Waterloo, Canada, 1970.
9. Floyd,R.W., *Algorithm 97, shortest path*, Comm. ACM. Vol.5, No.6, P.345, 1962.
10. Ford,L.R.,Jr., *Network flow theory*, Report P-923, Rand Corp., Canada, 1956.
11. Glover,F., D.Klingman, N.Philips, and R.F.Schneider, *New polynomial shortest path algorithms and their computational attributes*, Man. Sci., Vol.31, P.1106-1128, 1985.
12. Kershenbaum,A., *A note on finding shortest path trees*, Networks, Vol.11, 399-400, 1981.
13. Moore,E.F., *The shortest path through a maze*, in: Proceedings of the Internatioal Symposium on the theory of switching Part II; The Annals of the computation Laboratory of Havard University 30, Havard University Press, Boston, U.S., 1957.