지체호를 사용하는 시간 전개형 네트워크의 개발

이달상* · 김만식** · 이영해**

Development of Time-Expanded Network using Hold-over Arcs

Dal-Sang Lee*, Man-Sik Kim** and Young-Hae Lee**

Abstract

The problem of scheduling the passage things with low transit priority to maximize the amount that can be sent during given time periods without interfering with the fixed schedule for passage things with high transit priority in a track, is treated in this paper.

The problem is transformed into the Time Expanded Network without traverse time through the Ford and Fulkerson Model and the Enumeration Algorithm is developed for solutions using TENET GENerator(TENETGEN).

Finally, the proposed algorithm is compared with Dinic's maximal-flow algorithm and examined for the availability of the procedures on the personal computer.

1. 서 론

출발지 O와 목적지 R를 연결하는 최소걸침나무(a minimum spanning tree)를 생각하자. 이 최소걸침나무에서는 출발지 O와 목적지 R 사이에 단 하나의 경로(path)만이 존재한다. 이때 출발지에서 목적지로 가는 여러 종류의 통과물이 있으며 이러한 여러 종류의 통과물이 경로상의 각 호(arc)를 통과하는 시간은 각각 다를 수 있다. 또한 통과물들의 우선순위가 정해져 있어서 통과 우선순위가 높은 통과물이 한 호를 통과할 시에는 통과 우선순위가 낮은 통과물은 통과 우선순위가 높은 통과물이 통과할 때까지 마디(node)에 머물러 있어야 하며 이를 위해 각 마디에는 통과물이 머물 수 있는 장소가 있으며 용량이 부과되어 있다. 또 각 통과물 사이에는 일정한 시간 간격이 유지되어야 한다. 한편 이 걸침나무에 있어서 다른 두 마디간의 경로들이 출발지 O와 목적지 R 사이의 본 경로의 일부를 사용할 수도 있으며 (그림 1), 이때는 본 경로보다 통과우선순위가 높다고 가정한다. 본 연구의 목적은 통과우선순위가 높은 통과물의 경로 통과 일정이 알려져 있을 때 통과우선순위가 낮은 통과물의 최

* 동의대학교 산업공학과
** 한양대학교 산업공학과

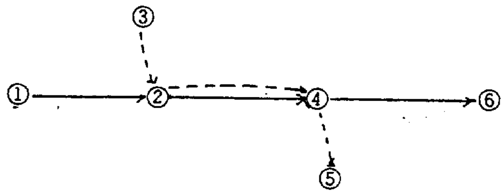

그림 1. 다른 경로의 일부로 사용되는 경로

대흐름량과 그때의 일정을 결정하는 것이다. 이러한 문제가 적용되는 분야로서는 차량, 항공기, 열차의 스케줄링 문제[10]를 들 수 있다.

지체호를 사용하는 시간 전개형 네트워크는 다음 문제를 해결하기 위해 개발되어진다. 출발지 O와 목적지 R중간에 s개의 마디가 있어, 통과 우선순위가 높은 통과물이 통과하게 될 때는 통과 우선순위가 낮은 통과물이 비켜줄 수 있도록 대피할 수 있는 장소가 마련되어 있다. 또한 경로의 일부분은 다른 경로의 일부로 사용될 수 있으며, 이 경우 그 구간을 통과하는 통과물 시각은 정해져 있으며 통과 우선순위가 원 경로보다 높은 것으로 간주한다. 이때 출발지 O와 목적지 R까지, 현재 통과 우선순위가 높은 통과물의 통과물 시각표를 변경시키지 않으면서 일정 기간 동안 보낼 수 있는 통과 우선순위가 낮은 통과물의 최대 흐름량과 그때의 일정들을 구하는 문제이다. 이 문제는 동적 네트워크(Dynamic Network)의 최대 동적 유량(Maximum Dynamic Flow)문제로 변환시켜 풀 수 있다.

본 연구에서는 위 문제를 Ford-Fulkerson 모형[5]을 이용하여 이동시간이 제거된 시간 전개형 네트워크(Time-Expanded NETwork : TENET)로 변환하고 TENET Generator를 개발, 사용하여 TENET의 데이타를 구한다. 또한 변환된 TENET의 구조적 단순성과 방향성을 이용하여 해를 찾는 간단한 열거 해법을 제시하고 이 열거 해법과 Dinic의 최대 흐름 해법의 계산량(complexity)[6]을 비교함으로써 기존의 해법보다 우수함을 보인다.

2. 최대 동적 유량 문제와 시간 전개형 네트워크

동적 네트워크(dynamic network : DNET)란 호의 용량 및 비용 파라메타 이외에 호를 통과하기 위해 소요되는 시간, 즉 이동시간(travel time)이 있는 네트워크를 말한다. 동적 네트워크에서는 유량을 동적유량(dynamic flow)라고 하는데 최대 동적 유량 문제는 지정된 시간 P동안의 최대 동적 유량을 구하는 문제이다.

일반적으로 동적 유량 문제는 직접 동적 네트워크 G에서 최적화하는 방법[1]과 동적 네트워크 G를 이동 시간의 파라메타가 제거된 시간 전개형 네트워크 (TENET), G_T로 변환하여 이 변환된 네트워크에 정적 네트워크의 해법을 적용하여 최적해를 구하는 방법[2,5]이 있다.

이 중 TENET는 네트워크 구조(교점과 호 및 각 파라메타)가 시간적으로 변하는 복잡한 동적 네트워크도 쉽게 표현될 수 있으며 특히 각 통과물 간의 시간 간격을 일정한 시간 이상 유지해야 하는 단일 경로의 통과 우선순위가 있는 통과물 일정계획 문제는 동적 네트워크보다는 시간 전개형 네트워크가 다루기 쉽다.

DNET에서 TENET로 전개하는 과정은 다음과 같다.

G(N, A)를 마디 집합 N과 호 집합 A를 가진 유방향 네트워크라 하고 $|N|=n$ 그리고 $|A|=m$이라 하자. 각 호$(x, y)\in A$에 대해 $c(x, y)$는 비음정수인 용량(capacity)를 나타내고 $a(x, y)$는 이동 시간 (arc-travel time)을 뜻한다. 또한 각각 $|O|=|R|=q\geq 1$를 갖는 sources와 sinks인 집합 $O\subset N$과 $R\subset N$가 주어지고 $O=(o_1, o_2, \cdots, o_q)$와 $R=(r_1, r_2, \cdots, r_q)$라 하면 $i=1, 2, \cdots, q$에 대해 o_i에서 r_i로 들어가는 흐름만이 고려의 대상이 된다. 시간 $t=0, 1, 2, \cdots, P$에서 네트워크 G상에 흐름이 발생한다면 최대 동적 흐름문제는 P시간 동안에 목적지 $r_1, r_2, \cdots, r_q$에 도달하는 최대 흐름량과 이 흐름이 거치는 경로를 구하는 것이다. 이때 호의 용량과 이동 시간이 해에 영향을 주고 이것은 TENET로 가장 잘 나타 내어진다. TENET, $G_T(N_T, A_T)$는 다음과 같은 방법으로 구축된다. 각 마디 $x\in N$에 대해 $t=0, 1, 2, \cdots, P$에서 마디 $x(t)$를 정의하고 각 호$(x, y)\in A$에 대해 $t=0, 1, 2, \cdots, P-a(x, y)$에서 용량이 $c(x, y)$ 호$(x(t), y[t+a(x, y)])$를 구한다. 마지막으로 $x\in O\cup R$인 모든 x에 대해, $t=0, 1, 2, \cdots, P-1$에서 호$(x(t), x(t+1))$을 부가하고 거기에 무한대의 용량을 할당한다.

그러면 네트워크 G_T는

$|N_T|=(P+1)\times n$이고

$|A_T|=\Sigma_{(x,\ y)\in A \text{ and } p\geq a(x,\ y)}\ [P+1-a(x, y)]$
$+P(\ |O|+|R|\ )$

이다. 이때 P가 충분히 크면, 모든 $(x, y)\in A$에 대해 $P-a(x, y)\geq 0$이고, 따라서

$|A|=m(P+1)-\Sigma_{(x,\ y)\in A}\ a(x, y)+2Pq$

이 경우 최대 동적흐름 문제는 G에서 $i=1, 2, \cdots, q$에 대해 $o_i(0)$에서 $r_i(P)$로 가는 최대 흐름량을

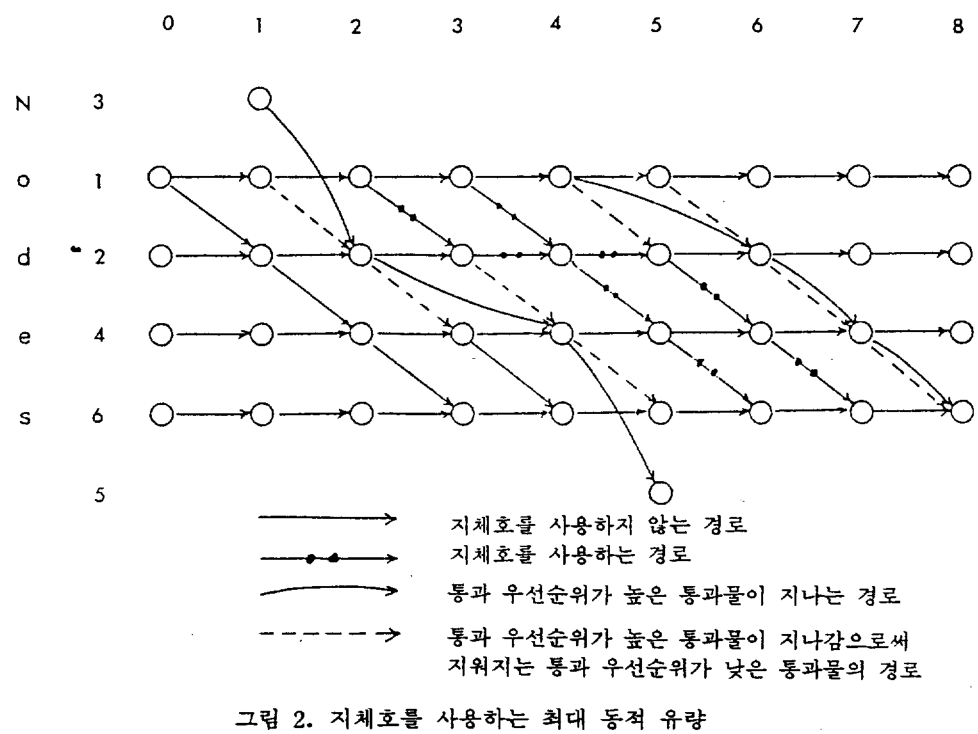

그림 2. 지체호를 사용하는 최대 동적 유량

구하는 것이다. 이에 대한 연구로서 Ford and Fulkerson[5]은 주어진 P에 대해 G를 G_T로 변환하여 중개 수송문제 (transshipment problem)를 풀어 해를 구하였고 Minieka[8]는 Ford and Fulkerson 알고리즘을 수정하여 최지출발－최조도착 (a latest-departure earliest-arrival) 최대 동적흐름 모형을 구축하여 해를 구하였다. Orlin[11]은 기간당 평균비용을 최소화하는 무한 기간의 동적유량문제 (infinite horizon dynamic flow problem)를 다루었고, 또한 Orlin[10]은 최대 동적 네트워크 유량문제의 특수한 경우로서 고정된 정기 일정을 만족하는 차량의 수를 최소화하는 문제를 다루었다. 그러나 이들 연구의 이동시간이 제거된 시간 전개형 네트워크에서는 지체를 피하는 최대 동적 유량이 항상 존재하기 때문에 이 지체호에 부과된 용량은 중요하지 않았다. 그러나 통과 우선순위가 높은 통과물에 의해 기존의 시간 전개형 네트워크가 수정되면 지체호를 사용해야 하는 최대 동적 유량이 존재하게 되며 그림 1. 에서 나타날 수 있는 한 예가 그림 2. 에 나타나 있다.

그림 2. 에서 최대 유량을 구하려는 경로는 1→2→4→6이고 기존 통과물이 각 마디 사이를 이동하는 시간은 1시간이며 마디 2와 마디 4는 3→2→4→5의 일부로 쓰여질 수 있다. 또한 통과 우선 순위가 높은 통과물이 4시 부터 마디 1과 마디 6사이의 경로를 사용함을 나타낸다. 여기서 지체호를 사용하지 않는 경로는 1개, 지체호를 사용하는 경로가 2개로 총 3개의 경로를 갖는다.

3. 단일 경로의 최대 흐름 스케쥴링을 위한 TENET 모형

3-1. 기호 설명

T : 지정된 시간 P

d : 각 통과물이 유지해야 하는 최소한의 시간 간격

s : 출발지 O와 도착지 R사이의 마디의 수

n_i : 마디 i가 대피시킬 수 있는 통과 우선순위가 낮은 통과물의 수

(i=1, 2, 3, …, s+2)

$t_i^{(k)}$: 통과물 k의 마디 i와 마디 i+1사이의 이동 시간

k=1 : 통과 우선순위가 높은 통과물

k=2 : 통과 우선순위가 낮은 통과물

3-2. 가　정

1) 출발지 O와 목적지 R는 한개의 경로로 연결되어 있다.
   즉 O에서 R로 가는 경로는 하나 뿐이다.
2) 각 통과물의 마디에서 출발과 도착시간 간격은 안전을 위해 d시간 이상이어야 한다.
3) 통과물 k의 지점 s와 지점 s+1까지 이동시간 T는 d의 정수배이다.
4) 각 통과물이 출발지 O를 떠나는 시간 역시 d의 정수배이다.
5) 통과 우선 순위가 높은 통과물이 한 지점을 통과할 때는 그 지점을 통과하는 우선순위가 낮은 통과물은 대피하여 그 통과물이 통과할 때까지 기다려야 한다.
6) 각 마디는 용량을 갖는다.

출발지는 마디 1로 도착지를 마디 s+2로 번호를 붙이면 마디⟨i, j⟩는 시각 j에서의 마디 i(i=1, 2, …, s, s+1, s+2 그리고 j=0, d, 2d, …, T/d)를 나타낸다. 특히 마디⟨1, 0⟩은 출발지, 마디⟨s+2, T/d⟩는 최종 도착지를 의미한다.

오직 한 통과물(k=2)이 시각 j에서 마디 i를 떠나서 마디 i+1에 도착하기 전 k=1인 통과물에 의해 추월되지 않을 경우에만 마디⟨i, j⟩와 마디⟨i+1, j+$t_i^{(2)}$⟩에 용량인 1인 호 (⟨i, j⟩, ⟨i+1, j+$t_i^{(2)}$⟩)가 존재하게 된다. 또한 j=0, d, 2d, …, T/d에 대해 마디⟨i, j⟩와 마디⟨i, j+1⟩ 사이에 용량이 n_i인 호

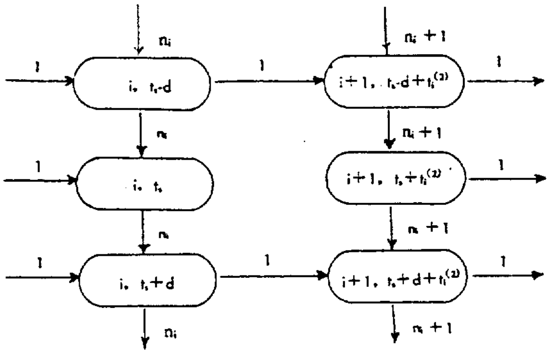

그림 3. TENET의 일부 구조

($\langle i, j\rangle$, $\langle i, j+1\rangle$)이 존재하며 이외의 다른 호는 존재하지 않는다. 예를 들어 $k=1$인 통과물이 마디 i를 출발하여 마디 $i+1$에 닿기 전에 이동시간 $t_i^{(2)}$를 가진 $k=2$인 통과물을 추월할 수 있다면 그림 3.은 TENET의 일부분을 표현하게 된다.

이러한 구조를 수리 모형으로 작성하면 목적함수는 마디$\langle 1, 0\rangle$에서 마디$\langle s+2, T/d\rangle$까지의 최대 흐름량을 구하는 것이 된다.

3-3. 수리모형

$G_T(N_T, A_T)$에서 통과 우선순위가 높은 통과물에 의해 호가 삭제되고 지체호가 추가됨으로 수정된 유한 유방향 그래프를 $G_T'(N_T', A_T')$라 하면 최대 유량에 대한 식 구성은 다음과 같다.

Maximize V

Subject to

$$\Sigma_{\langle u, v\rangle \in N_T'} X_{\langle l, m\rangle\langle u, v\rangle} - \Sigma_{\langle w, z\rangle \in N_T'} X_{\langle w, z\rangle\langle l, m\rangle} = \begin{cases} V & \text{if } \langle l, m\rangle = O \\ 0 & \text{if } \langle l, m\rangle \neq O,\ R \\ & \quad \text{for all } \langle l, m\rangle \in N_T' \\ -V & \text{if } \langle l, m\rangle = R \end{cases}$$

각 호($\langle l, m\rangle$, $\langle u, v\rangle$) A_T'에 대해

$$0 \leq X_{\langle l, m\rangle\langle u, v\rangle} \leq C_{\langle l, m\rangle\langle u, v\rangle}$$

여기서

$\langle 1, m\rangle$ $\langle u, v\rangle$, $\langle w, z\rangle$은 마디를 표시

$1, u, v \in I=\{I : I=1, 2, \cdots, s+2\}$

$m, v, z \in J=\{J : J=0, d, 2d, \cdots, T/d\}$

N_T' : 수정된 그래프 G_T'의 모든 마디(node)의 집합

A_T' : 수정된 그래프 G_T'의 모든 호(arc)의 집합

V : 구하고자 하는 총 유량

X_{ij} : 호(i, j)의 흐름량

C_{ij} : 호(i, j)의 용량

4. 시간 전개형 네트워크 Generator (TENETGEN)

DENT G를 TENET G_T로 변환 할때 지정된 시간 T가 커지면 변환 네트워크 G_T는 구조는 복잡하지 않으나 크기가 커지는 단점이 있다. 예를들어 출발지와 도착지를 포함하여 통과 지점이 20개이고 지정기간 T가 24시간, 시간간격 $d=0.1$일 경우 총 마디 수가 4800개 arc 수의 상한이 9575개인 네트워크로 된다. 이처럼 큰 네트워크를 수작업으로 입력시킨다는 것은 매우 불편한 일이며, 따라서 데이타를 간단히 입력시킬 수 있는 방법을 구하게 된다. 이에 TENET의 호 방향의 규칙성을 이용하여 TENETGEN이라고 명명한 TENET 발생기를 개발한다. TENET 발생기를 개발하기 위해 통과 우선순위가 있는 경로의 최대흐름 문제에 있어 TENET의 구조적 규칙성의 일부를 간략히 그림으로 표시하면 그림 4, 5, 6, 7과 같다.

그림 4.는 마디 i와 마디 $i+1$를 통과하는데 걸리는 시간이 통과 우선순위가 높은 통과물과 통과 우선순위가 낮은 통과물보다 더 느릴 경우를, 그림

5.는 통과 우선순위가 높은 통과물이 통과 우선순위가 낮은 통과물보다 더 빠를 경우, TENET-GEN에서 일어나는 호의 삭제를 나타낸다. 이들 경우 통과 우선순위가 높은 통과물에 대한 호의 시점에서 나가는 호의 종점으로 들어오는 호들 중 지체호를 제외한 모든 호가 삭제된다. 그림 6.은 주어진 기간 T=4가 단속적으로 일어난 경우에 생성되는 TENET의 형태이다. 이 경우 출발지 ⟨1, 0⟩에서 목적지 ⟨3, 4⟩까지의 경로 수는 0개이다. 즉 4시간 동안 출발지 1에서 목적지 3까지 갈 수 있는 통과 우선순위가 낮은 통과물의 수는 0이다. 그림 7.은 T=4가 계속 연속 된다고 할 때 생성되는 TENET의 형태를 나타낸다. 이 경우 출발지 ⟨1, 0⟩에서 목적지 ⟨3, 3⟩까지 생성되는 경로의 수는 2개이고 그 경로는 ⟨1, 2⟩→⟨2, 3⟩→⟨3, 1⟩, ⟨1, 3⟩→⟨2, 0⟩→⟨3, 2⟩이다. 즉 통과 우선순위가 높은 통과물의 시각표를 변경시키지 않으면서 T=4시간동안 통과 우선순위가 낮은 통과물을 보낼 수 있는 최대의 수는 2대이고 그때 출발지의 출발시각은 2시, 3시이다.

TENET의 기본적인 개념은 각 마디 사이에 통과 우선순위가 낮은 통과물의 진행 호를 발생시켜 놓고, 통과 우선순위가 높은 통과물의 진행 호가 발

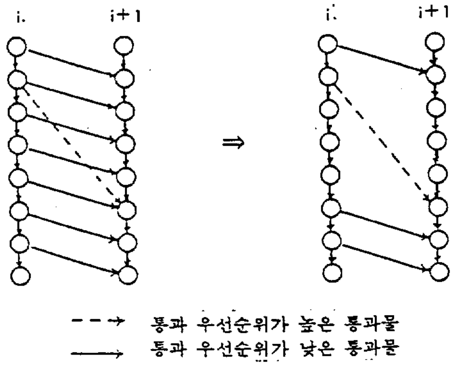

그림 4. $t_i^{(1)}>t_i^{(2)}$인 경우의 호의 삭제

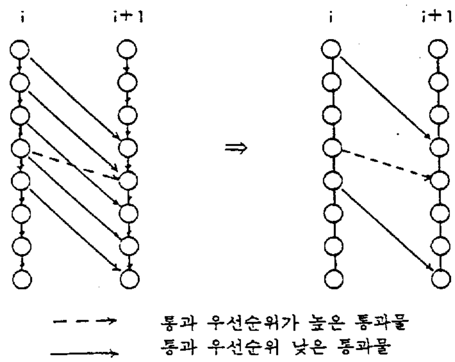

그림 5. $t_i^{(1)}<t_i^{(2)}$인 경우의 호의 삭제

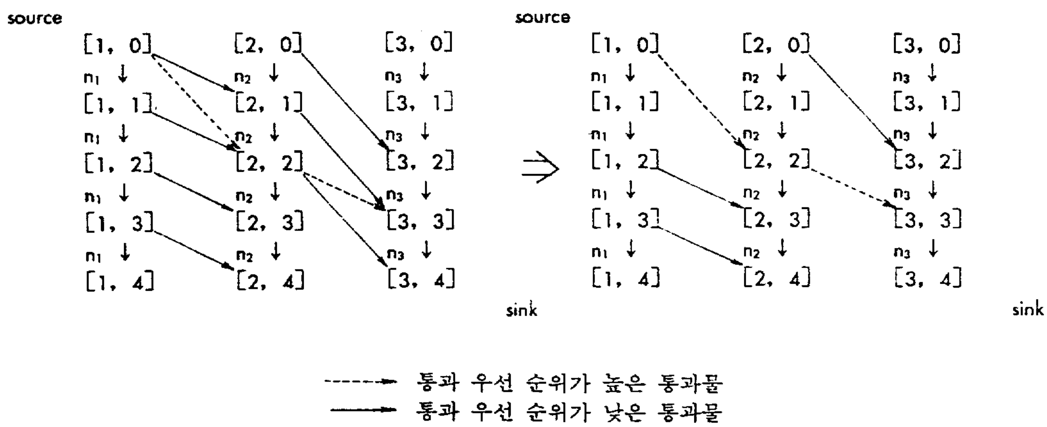

그림 6. Acylic TENET의 생성구조(s=1, T=4, d=1)

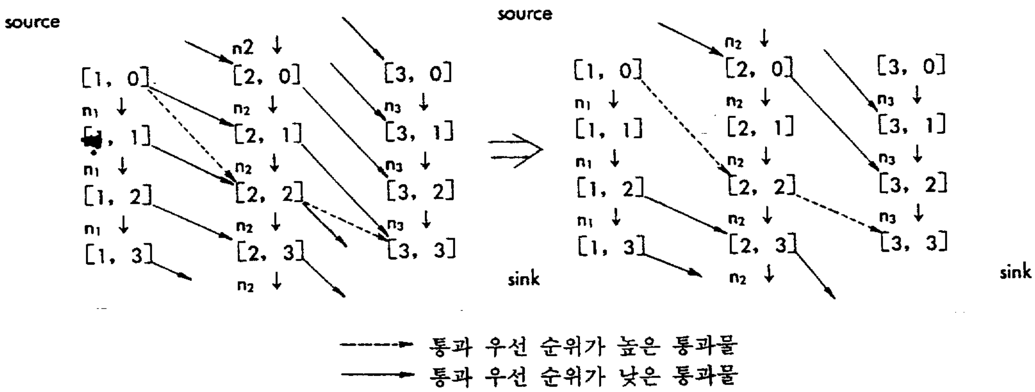

그림 7. Cyclic TENET의 생성구조(s=1, T=4, d=1)

생될때 TENET 구조의 규칙성을 따라 이미 생성된 통과 우선순위가 낮은 통과물의 진행 호를 소거해 가는 것이다. 여기에서 생성된 시간전개형 네트워크(NETET)는 각 마디(node)에서 나오고 들어가는 호는 각각 최대 2개 이고 종점(sink)으로 향하는 단방향의 특성을 갖고 있다.

5. 알고리듬

시간전개형 네트워크를 이용한 경로의 최대 흐름 스케쥴링을 구하는 절차는 다음과 같다.

A : TENETGEN을 이용한 TENET의 발생

단계 1 : 중간마디의 수(s), 계획기간(T), 시간간격(d). TENET의 종류(1 : Acyclic Net, 2 : Cycle Net)를 입력

단계 2 : 각 마디에서의 용량(n_i)를 입력(i=1, 2,⋯, s+2)

단계 3 : 통과 우선순위가 높은 통과물의 통과 우선순위가 낮은 통과물의 각 마디간 운행시간 $T_i^{(1)}$, $T_i^{(2)}$를 입력(i=1, 2,⋯, s+1)

단계 4 : 통과 우선순위가 높은 통과물의 출발지에서의 발차시각 (PST)을 입력

단계 5 : TENETGEN을 실행

B : 최대 흐름 스케쥴링 결정

A의 단계 5에서 발생된 데이타로 다음의 열거 해법을 사용하여 해를 구함.

단계 1

(1. 1) 출발지에서 우선 순위가 낮은 통과물의 출발 시각에 대한 조사되지 않은 첫 마디에서 다음 목적지로 나가는 호가 있는지를 조사한다.

(1. 2) 있으면 이 마디에 표지하고 이 호의 목적지를 마디 〈i, j〉라 하면 이 호의 용량을 1만큼 감소시키고 단계 2로 가고, 없으면 (1. 1)로 간다.

(1. 3) 출발지에서 마디가 모두 조사되었으면 끝낸다.

단계 2

(2. 1) 마디 〈i, j〉가 최종 목적지이면 단계 3으로 간다.

(2. 2) 마디 〈i, j〉가 최종 목적지가 아니면 마디 〈i, j〉에서 마디 〈i+1, j+$t_i^{(2)}$〉로 가는 호가 있는지를 조사한다.

(2. 1. 1) 만약 있으면 마디 〈i, j〉에 표지하고 마디 〈i+1, j+$t_i^{(2)}$〉로 가는 호의 용량을 1 만큼 감소시키고 i=i+1, j=j+$t_i^{(2)}$로 놓고 (2. 1)로 간다.

(2. 2. 2) 없으면 마디 〈i, j〉에 표지하고 마디 〈i, j+d〉로 가는 호가 있는지를 조사한다.

(2. 2. 2. 1) 있으면 마디 〈i, j+d〉로 가는 호의

용량을 1 만큼 감소시키고 j=j+d로 놓고 (2. 1)로 간다.

(2. 2. 2. 2) 없으면 지금까지 표지된 경로의 용량을 1 만큼 증가시킴과 아울러 표지된 것을 모두 지우고 단계 1로 간다.

단계 3

(3. 1) 경로의 수를 1 만큼 증가시키고 지금까지의 표지를 모두 지운 후 단계 1로 간다.

6. 실험 결과 및 비교

6-1. 계산량(complexity)의 비교

생성된 시간 전개형 네트워크(TENET)에서 구조적 단순성과 단방향성을 이용하여 경로들을 살펴보면 출발지에서 목적지로 가는 최대 경로의 수는 T/d개이다. 이것은 각 마디간 절단의 최대 크기를 의미한다. 또한 각 경로는 최소 s+1개, 최대 (s+1+T/d)개의 호로 구성된다. 따라서 열거 해법의 최악의 계산량은 (T/d)*(s+1+T/d)이며 O(N)으로 표시할 수 있다. 이것은 Dinic 해법의 complexity, $O(N^2M)$, Karzanov해법의 complexity, $O(N^3)$ [6, 9]보다 작다. 이러한 이유는 위의 정교한 해법들이 일반적인 복잡한 네트워크에 대해 효율적 해를 구하기 위해 추가로 계산을 더하기 때문이다.

6-2. 실험의 시행과 결과

본 연구에서 계산 실험을 위하여 사용한 컴퓨터는 IBM PC/AT 호환기종을 사용하였다. 한편 프로그램의 컴파일 및 링크과정에는 Microsoft FORTRAN 77 Ver. 3.3 Compiler와 Microsoft 8086 Object Linker V. 3.04를 사용하였으며 DOS V.3.2 하에서 시행하였다. 여기서 계산시간은 Macro-Assembler로 부프로그램을 작성하여 시스템 시간을 측정하였으며, 이 때 입출력에 소요되는 시간은 배제하였다. 실험을 시행한 결과가 표 1.에 요약되어 있다.

위 실험에서는 본 연구에서 개발한 열거 해법과 Dinic의 최대 유량 해법을 비교, 검토하기 위하여

표 1. 실험의 결과

No	S	단위 : 시간		Network의 크기		우선순위 통과물의 발생수	Network의 종류	생성된 경로의 수	실행시간(초)	
		T	d	N	M				Dinic해법	열거해법
1	1	5.0	1.0	18	20	1	A	0	0.00	0.00
				15	20		C	3	0.05	0.00
2	3	12.0	0.1	605	1017	6	A	89	4.56	0.22
				600	1018		C	96	5.88	0.22
3	5	12.0	0.1	847	1468	6	A	87	4.84	0.27
				840	1468		C	96	6.54	0.33
4	5	24.0	0.1	1687	2956	11	A	190	16.97	0.61
				1680	2953		C	196	19.98	0.72
5	10	12.0	0.1	1452	2633	6	A	87	5.93	0.33
				1440	2620		C	102	8.29	0.33
6	10	24.0	0.1	2892	5290	11	A	192	19.00	0.59
				2880	5265		C	207	23.84	0.60
7	15	12.0	0.1	2057	3764	6	A	69	6.54	0.44
				2040	3718		C	96	10.48	0.49

표 1. 실험의 결과(계속)

No	S	단위 : 시간		Network의 크기		우선순위 통과물의 발생수	Network의 종류	생성된 경로의 수	실행시간(초)	
		T	d	N	M				Dinic해법	열거해법
8	15	24.0	0.1	4097	7540	11	A	169	20.38	0.98
				4080	7478		C	196	28.45	1.04
9	21	24.0	0.1	5543	10357	11	A	154	19.83	0.88
				5520	10303		C	196	30.31	0.88
10	25	24.0	0.1	6507	12221	11	A	147	21.80	1.21
				6480	12113		C	196	34.32	1.21
11	31	24.0	0.1	7953	14939	11	A	130	24.83	1.81
				7920	14751		C	196	43.50	2.13

S : 중간 node의 수 T : Time periods d : 통과물 사이의 시간 간격 N : node 수 M : arc 수.

PC에서 실행할 때 다룰 수 있는 최대 크기의 네트워크를 주로 다루었으며, 각 마디에서 통과물를 대피시킬 수 있는 장소의 크기는 통과물의 3~5배 사이에서, 각 통과물의 각 마디간 이동시간은 b의 3~4배 내에서, 우선 순위가 높은 통과물의 발생수는 T/b의 4.5%에서 임의로 주었으며, 이 때 사용된 난수는 IMSL의 난수 발생 SUBROUTINE GGUBS를 이용하여 발생시켰다.

또한 실험은 같은 데이타를 가지고 Acyclic Net와 Cyclic Net 두 경우 모두를 행하였는 바 Cyclic Net의 경우 생성된 경로의 수가 많았고, 각 마디간의 이동시간이 작을 경우 더 많은 경로가 발생하였는데 이것은 당연한 결과이다.

표 1.에서 보이듯 열거 해법이 Dinic 해법보다 20~30배 정도 해를 구하는 속도가 빨랐다. 위 실험결과 중 최대 실행시간은 Dinic 해법에서 43.50초인 반면 열거 해법은 2.13초로 20배 정도 작고, 시간의 절대값이 작으므로 본 열거 해법의 PC에서의 이용 가능성을 보여준 것이라 하겠다.

7. 결 론

본 연구에서는 경로의 최대 흐름 일정을 결정하는데 시간 전개형 네트워크(TENET) 모형과 발생기(TENETGEN)를 개발, 사용하고 여기에서 얻어진 네트워크의 구조적 단순성과 방향성을 이용하여 열거해법을 개발하였다. 이 열거 해법과 Dinic의 최대 흐름 해법의 계산량(complexity)을 비교검토하고 PC의 이용 가능성에 대한 검토를 위해 실험을 해 본 결과, 열거 해법이 Dinic의 해법보다 20~30배 정도 빠른 결과치를 얻었다.

이것은 Dinic 해법이 네트워크의 구조와는 무관하게 최대 유량을 구하기 위해 층(layer) 네트워크를 구성하는 바, 이에 소비되는 시간이 많았기 때문이라 사료된다. 또한 본 열거 해법에서 차지하는 기억용량은 Dinic 해법보다 50% 이상 작아서 훨씬 더 큰 규모의 문제를 다룰 수 있다.

앞으로 PC의 성능이 더욱 향상됨에 따라 TENET의 이용 가능성은 더욱 커질것이고 아울러 본 열거 해법의 효용도 크리라 기대된다.

참고문헌

[1] 강맹규, 네트워크와 앨고리즘, 한양대, 1983.

[2] 이달상, 김만식, "시간 전개형 네트워크를 이용한 선로의 최대흐름 스케쥴링," 대한교통학회

지, 제8권 2호, pp. 67-75, 1990.

[3] Bellmore, M. and R.R. Vemuganti, "On Multi-Commodity Maximal Dynamic Flows," *Oper. Res.*, Vol. 21, No. 1, pp. 10-21, 1973.

[4] Ford, L.R., Jr. and D.R. Fulkerson, "Constructing Maximal Dynamic Flows from Static Flows," *Oper. Res.*, Vol. 6, pp. 419-433, 1958.

[5] Ford, L.R., Jr. and D.R. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N.J., 1962.

[6] Goldfarb, D. and M.D. Grigoriadis, "A Computational comparison of the Dinic and Network Simplex Methods for Maximum Flow," *Annals of Oper. Res.*, Vol. 13, pp. 83-123, 1988.

[7] Hillier, F.S and G.J. Lieberman, *Introduction to Operations research, 5th* Edition, McGraw-Hill Pub. Co., Singapore, 1990.

[8] Minieka, E., "Maximal, Lexicographic, and Dynamic Network Flows," *Oper. Res.*, Vol. 21, No. 2. pp. 517-527. 1973.

[9] Nemhauser, G.L. and A.H.G. Rinnooy Kan, *Handbooks in Operations Research and Management Science : Optimization*, Vol. 1, Chap. IV, North-Holland, 1989.

[10] Orlin, J.B., "Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule : An Application of Periodic Posets," *Oper. Res.*, Vol. 30, pp. 760-776, 1982.

[11] Orlin, J.B., "Maximum-Throughput Dynamic Network Flows," *Math. Prog.*, Vol. 27, pp. 214-231, 1983.

[12] Wilkinson, W.L., "An Algorithm for Universal Maximal Dynamic Flows in a Network," *Oper. Res.*, Vol. 19, pp. 1602-1612, 1971.