

통과 우선순위가 있는 선로의 최대 흐름문제 —Maximal-Flow-Problem with transit priority in a track—

이 달 상*
김 만 식**

ABSTRACT

This paper treats the problem to schedule for train with low transit priority so as to maximizing the number that can be sent during given time without interfering with the fixed schedule for train with high transit priority in a track.

We transform the this problem into Time-Expanded Network without traverse time through application of Ford-Fulkerson Model, develop a TENET GENerator(TENETGEN) and obtain the data of TENET using developed TENETGEN.

Finally, we seek the optimal solution to these data with Dinic's Maximal-Flow Algorithm and examine the availability of our procedures in personal computer.

I. 서 언

철도로 연결되어 있는 출발지 S와 목적지 D 사이의 물동량이 크게 증대되면, 열차 시작표의 조정 혹은 선로의 증설에 따른 검토가 필요하게 된다. 특히 막대한 비용이 드는 선로의 증설을 검토하게 될 때는 필수적으로 현재 기존 선로의 최대 흐름용량을 검토하게 된다. 그러나 통과 우선순위가 다른 열차의 종류가 많은 경우, 열차 시작표의 작성은 물론 기존 선로의 최대흐름 용량을 파악하는 것은 쉽지 않다.

기존선로의 최대 흐름 용량을 결정하는 중요한 인자로서는 통과 우선 순위가 다른 열차의 수, 각 역과 역사이의 운행시간, 안전을 위해 필수적인 각 열차 사이의 최소한의 시간 간격 등이 있다. 통과 우선 순위를 가진 선로의 최대 흐름량 결정이란 다음과 같다.

출발지 S와 목적지 D 중간에 S개의 역 혹은 간이역이 설치되어 있어, 통과 우선 순위가 높은 열차가 통과하게 될 때는 통과 우선 순위가 낮은 열차가 비켜 줄 수 있도록 대피할 수 있는 장소가 마련되어 있다. 이 경우 출발지 S와 목적지 D까지, 현재 통과 우선 순위가 높은 열차의 열차 시작표를 변경시키지 않으면서 일정기간 동안 보낼 수 있는 통과 우선 순위가 낮은 열차의 최대 흐름량을 구하는 문제이다. 이 문제는 전형적인 동적 네트워크(Dynamic Network)의 최대동적 유량 문제이다.

본 연구에서는 위 문제를 Ford-Fulkerson 모형을 응용하여 이동시간이 제거된 시간 전개형 네트워크(Time-Expanded NETwork, TENET)로 변환하고 TENET Generator를 개발, 사용하여 TENET의 data를 구한다.

이 data를 이용하여 Dinic의 최대 흐름 해법으로 최적해를 구하며 이 절차의 PC에서의 실용 가능성을 검토한다.

* 동의대학교 산업공학과 부교수

**한양대학교 산업공학과 교수

접수 1990년 4월 25일

2. 최대 동적 유량 문제와 시간 전개형 네트워크(Maximal-Dynamic-Flow Problem : MDFP and Time-Expanded Network : TENET)

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

일반적으로 동적 유량 문제를 직접 동적 network(G)에서 최적화하는 방법 [1]과 동적 network G를 이동 시간의 파라메타가 제거된 시간 전개형 network(TENET), G_T 로 변환하여 이 변환된 network에 정적 network의 해법을 적용하여 최적해를 구하는 방법 [2,3]이 있다.

이 중 TENET는 network 구조(교점과 호 및 각 파라메타)가 시간적으로 변하는 복잡한 동적 network도 쉽게 표현될 수 있으며 특히 각 열차 간의 시간 간격을 일정한 시간 이상 유지해야 하는 철도의 열차 scheduling 문제는 동적 network 보다는 시간 전개형 network이 다루기 쉽다.

DNET에서 TENET로 전개하는 과정은 다음과 같다.

$G=(N,A)$ 를 node 집합 N과 arc 집합 A를 가진 유방향 network라 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 $|T|=q>1$ 를 갖는 the sources와 the sinks인 집합 SCN과 TCN가 주어지고 $S=(S_1, S_2, \dots, S_q)$ 와 $T=(T_1, t_2, \dots, t_q)$ 라 하면 $i=1, 2, \dots, q$ 에 대해 S_i 에서 T_i 로 들어가는 흐름만이 고려의 대상이 된다.

시간 $t=0, 1, 2, \dots, p$ 에서 the network G상에 흐름이 발생한다면 최대 동적 흐름 문제는 p시간 동안에 the sinks t_1, t_2, \dots, t_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 node $x \in N$ 에 대해 $t=0, 1, 2, \dots, p$ 에서 node $x(t)$ 를 정의하고 각 arc $(x,y) \in A$ 에 대해 $t=0, 1, 2, \dots, p-a(x,y)$ 에서 용량이 $c(x,y)$ 인 arc $(x(t), y[t+a(x,y)])$ 를 구한다.

마지막으로 $x \in S \cup T$ 인 모든 x에 대해, $t=0, 1, \dots, p-1$ 에서 arc $(x(t), x(t+1))$ 을 부가하고 거기에 무한대의 용량을 할당한다.

그러면 network G_T 는

$$|N_T|=(P+1) * n \text{이고}$$

$$|A_T|=\sum_{(x,y) \in A \text{ and } p-a(x,y) \geq v} [p+1-a(x,y)]+p(|S|+|T|)$$

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

$$|A_T|=m(p+1)-\sum_{(x,y) \in A} a(x,y)+2pq$$

이 경우 최대 동적흐름 문제는 G_T 에서 $i=1, 2, \dots, q$ 에 대해 $S_i(0)$ 에서 $t_i(p)$ 로 가는 최대 흐름량을 구하는 것이다.

이에 대한 연구로서 Ford and Fulkerson [7]은 주어진 P에 대해 G를 G_T 로 변환하여 transshipment problem 을 풀어 해를 구하였고 Minieta [3]는 Ford and Fulkerson 알고리즘을 수정하여 a latest-departure earliest-arrival 최대 동적흐름 모형을 구축하여 해를 구하였다.

3. 통과 우선순위가 있는 선로 문제의 TENET 모형

3.1 가정

- I. 출발지 S와 목적지 D는 한 개의 선로로 되어 있다. 즉 S에서 T로 가는 경로는 하나 뿐이다.
- II. 각 열차의 역에서 출발과 도착시간 간격은 안전을 위해 d시간 이상이어야 한다.

- Ⅲ. 열차 k의 지점 s와 지점 s+1까지 이동시간 T는 d의 정수배이다.
- Ⅳ. 각 열차가 출발지 S를 떠나는 시간 역의 d의 정수배이다.
- Ⅴ. 통과 우선 순위가 높은 열차가 한 지점을 통과할 때는 그 지점을 통과하는 우선 순위가 낮은 열차는 제외하여 그 열차가 통과할 때까지 기다려야 한다.

3.2 기호 설명

- T: 지정된 시간
- d: 각 열차가 유지해야 하는 최소한의 시간 간격
- s: 출발지 S와 도착지 T 사이의 역의 수
- n_i : 역 i가 대피시킬 수 있는 통과 우선 순위가 낮은 열차의 수
($i=1, 2, 3, \dots, s+2$)
- $t_i^{(k)}$: 열차 k의 역 i와 역 i+1 사이 이동시간
- k=1: 통과 우선순위가 높은 열차
- k=2: 통과 우선순위가 낮은 열차

출발지는 역 1로 도착지를 역 s+2로 하면 node(i,j)는 시간 j에서의 역 i($i=1, 2, \dots, s, s+1, s+2$ and $j=0, d, 2d, \dots, T/d$)를 나타낸다. 특히 node(1,0)은 the source, node(s+2, T/d)는 the sink를 의미한다.

오직 한 열차(K=2)가 시각 j에서 역 i를 떠나서 역 i+1에 도착하기 전 K=1인 열차에 의해 추월되지 않을 경우에만 node(i,j)와 node(i+1, j+t_i⁽²⁾)에 용량이 1인 arc((i,j), (i+1, j+t_i⁽²⁾))가 존재하게 된다. 또한 $j=0, d, 2d, \dots, T/d-1$ 에 대해 node(i,j)와 node(i, j+1) 사이에 용량이 n_i 인 arc((i,j), (i,j+1))이 존재하며 이외의 다른 arc는 존재하지 않는다.

예를 들어 K=1인 열차가 역 i를 시각 t_s에 출발하여 역 i+1에 닿기 전에 이동시간 t_i⁽²⁾를 가진 K=2인 열차를 추월 할 수 있다면 그림 1.은 TENET의 일부를 표현하게 된다.

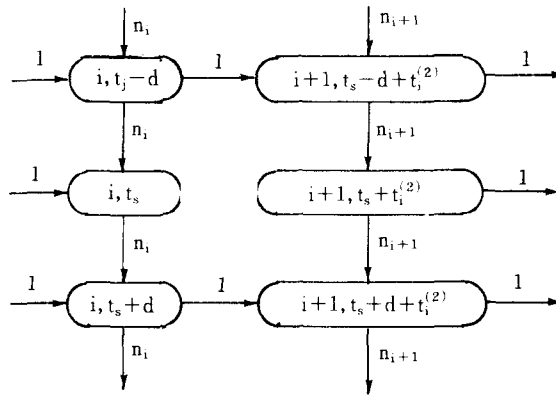


그림 1. TENET의 일부 구조

이러한 구조를 수리 모형으로 작성하면 목적함수는 node(1,0)에서 node(s+2, T/d)까지의 최대 흐름량을 구하는 것이 된다.

<수리모형>

Maximizing V

Subject to

$$\sum_{\langle u,v \rangle} x_{\langle l,m \rangle \langle u,v \rangle} - \sum_{\langle w,z \rangle} x_{\langle w,z \rangle \langle l,m \rangle} = \begin{cases} V & \langle l,m \rangle = S \\ 0 & \langle l,m \rangle \neq S, T \\ -V & \langle l,m \rangle = T \end{cases}$$

$$0 < x_{\langle l,m \rangle \langle u,v \rangle} < Cx_{\langle l,m \rangle \langle u,v \rangle}$$

여기서

$\langle l,m \rangle \langle u,v \rangle \langle w,z \rangle$ 은 node를 나타냄

$l, u, v \in I = \{I : I=1, 2, \dots, s-2\}$

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

V : 구하고자 하는 총 유량

x_{ij} : arc(i, j)의 흐름량

C_{ij} : arc(i, j)의 용량

4. 시간 전개형 네트워크 Generator(TENETGEN)

DGET G 를 TENET G_T 로 변환할 때 지정된 시간 T 가 커지면 변환 network G_T 는 구조는 복잡하지 않으나 크기가 커지는 단점이 있다.

예를들어 출발지와 도착지를 포함하여 통과 지점이 20개이고 지정기간 T 가 24시간, 시간간격 $d=0.1$ 일 경우 총 node 수가 4800개 arc 수의 상한이 9360개인 network로 된다. 이처럼 큰 network를 수작업으로 입력시키다는 것은 매우 불편한 일이며, 따라서 데이터를 간단히 입력시킬 수 있는 방법을 구하게 된다. 이에 TENET의 arc 방향의 규칙성을 이용하여 TENETGEN이라고 명명한 TENET Generator를 개발한다.

TENET Generator를 개발하기 위해 통과 우선 순위가 있는 선로의 최대흐름 문제에 있어 TENET의 구조의 규칙성의 일부를 간략히 그림으로 표시하면 그림 2., 그림 3.과 같다.

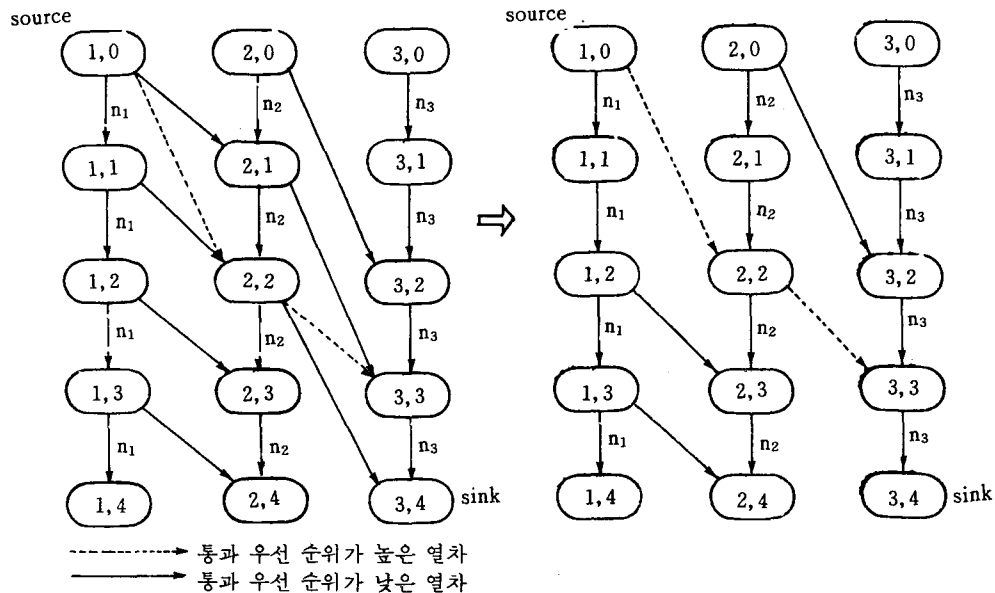


그림 2. Acyclic TENET 생성구조($s=1, t=4, d=1$)

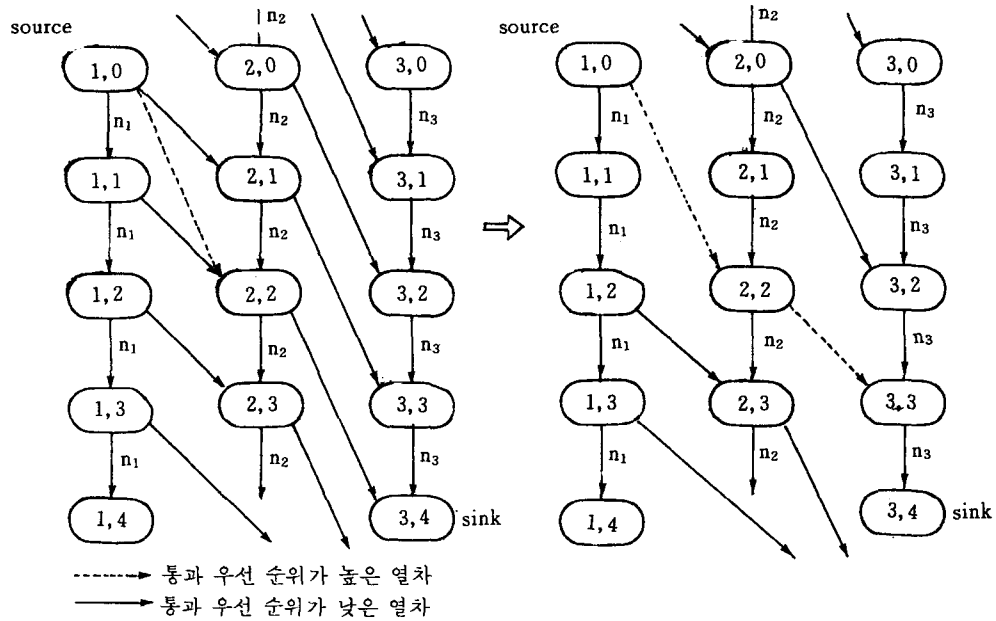


그림 3. Cyclic TENET 생성문제의 구조($s=1, t=4, d=1$)

그림 2는 주어진 기간 T 가 단속적으로 일어난 경우에 생성되는 TENET의 형태이다. 이 경우 source (1,0)에서 sink (3,4)까지의 경로 수는 0개이다.

즉 4시간 동안 출발지 1에서 목적지 3까지 갈 수 있는 통과 우선 순위가 낮은 열차의 수는 0이다.

그림 3은 T 가 계속 연속 된다고 할때 생성되는 TENET의 형태를 나타낸다. 이 경우 source(1,0)에서 sink (3,4)까지 생성되는 경로의 수는 2개이고 그 경로는 1,2→2,3→3,0, 1,3→2,4→3,1이다.

즉 통과 우선 순위가 높은 열차의 시각표를 변경시키지 않으면서 통과 우선 순위가 낮은 열차를 보낼 수 있는 최대의 수는 3대이고 그때 출발지의 출발시각은 2시, 3시, 4시이다.

TENETGEN의 기본적인 개념은 각 node 사이에 통과 우선 순위가 낮은 열차의 진행 arc를 발생시켜 놓고, 통과 우선 순위가 높은 열차의 진행 arc가 발생될 때 TENET 구조의 규칙성을 따라 이미 생성된 통과 우선 순위가 낮은 열차의 진행 arc를 소거해가는 것이다.

5. 통과 우선 순위를 가진 선로문제의 최대 동적 흐름량을 구하는 절차

A-1 : TENETGEN을 이용한 TENET의 발생

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

단계 2 : 각 역에서의 용량(n_i)를 입력 ($i=1, 2, \dots, s+2$)

단계 3 : 통과 우선 순위가 높은 열차와 통과 우선 순위가 낮은 열차의 각 역간 운행시간 $T1(i), T2(i)$ 를 입력 ($i=1, 2, \dots, s+1$)

단계 4 : 통과 우선 순위가 높은 열차의 출발지에서의 발차시간(PST)을 입력

단계 5 : TENETGEN을 실행

B-1 : 최대 흐름량 결정

A-1의 단계 5에서 발생된 DATA로 Dinic의 해법을 사용하여 해를 구함.

6. 실험의 시행과 결과

본 연구에서 계산 실험을 위하여 사용한 컴퓨터는 IBM PC/AT 호환기종을 사용하였다(Televideo At, 주기인 용량: 640K, clock속도: 12MHZ, 80287 coprocessor 채용).

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 system 시간을 측정하였으며, 이 때 입출력에 소요되는 시간은 배제하였다.

실험을 시행한 결과가 표 1에 요약되어 있다.

표 1. 실험의 결과

NO	S	단위 : 시간		Network의 크기		Network의 종류	생성된 PATH의 수	실행시간 (초)
		T	d	N	M			
1	3	4	1	25	27	A	0	0.05
	3	4	1	25	31	C	1	0.11
2	21	24	0.1	5280	9153	A	127	27.79
	21	24	0.1	5280	9313	C	167	62.01
3	31	24	0.1	7920	14802	A	94	31.53
	31	24	0.1	7920	13747	C	139	78.60
4	31	24	0.1	7920	14965	A	137	36.30
	31	24	0.1	7920	14283	C	169	67.84

S : 중간 역의 수 T : 지정 계획 시간 d : 열차 사이의 시간 간격

a : Acyclic Net, C : Cyclic Net

위 실험에서의 PC에서의 이용가능성을 검토하기 위하여 Dinic 해법을 PC에서 실행할 때 다룰 수 있는 최대 크기의 Network을 주로 다루었으며, 각 열차의 각 역간 이동시간은 d의 10배 내에서 임의로 주었다. 또한 실험은 같은 데이터를 가지고 Acyclic net와 Cyclic net 두 경우 모두를 행하였는 바 Cyclic Net의 경우 생성된 path의 수가 많았고, 각 역간의 이동시간이 작을 경우 더 많은 path가 발생하였는데 이것은 당연한 결과이다.

위 실험 결과 중 최대 실행 시간은 약 79초이었는데 이것은 TENET의 PC에서의 이용 가능성을 보여 준 것이라 하겠다.

7. 결 론

본 연구에서는 통과 우선순위가 있는 선로의 최대흐름량을 결정하는데 시간 전개형 network(TENET) 모형과 TENET Generator(TENETGEN)을 사용하고 PC의 이용 가능성에 대한 검토를 위해 실험을 해 본 결과 만족할 만한 결과를 얻었다.

또한 시간전개형 network(TENET)의 실용성을 제고시키기 위하여 TENETGEN이란 TENET Generator를 개발하였다.

앞으로 PC의 성능이 더욱 향상됨에 따라 TENET의 이용 가능성은 더욱 커질것으로 기대된다.

References

1. 강 맹규, 네트워크와 알고리즘, 한양대, 1988.
2. Ford, L. R., Jr. and D. R. Fulkerson, "Constructing Maximal Dynamic Flows from Static Flows," Opns. Res., Vol. 6, pp. 419~433, 1958.

3. Minieka, E., "Maximal, Lexicographic, and Dynamic Network Flows," *Opns. Res.*, Vol. 21, No. 2, pp. 517~527, 1973.
4. Hillier, F. S and G. J. Lieberman, *Introduction to Operations Research*, Fourth Edition, Holden-Day, Inc., Oakland, 1986.
5. Goldfarb, D. and M. D. Grigoriadis, "A computational Comparison of the Dinic and Network Simplex Methods for Maximum Flow," *Annals of Opns. Res.*, Vol. 13, pp. 83~123, 1989.
6. Bellmore, M. and R. R. Vemuganti, "On Multi-Commodity Maximal Dynamic Flows," *Opns. Res.*, Vol. 21, No. 1, pp. 10~21, 1973.
7. Ford, L. R., Jr. and Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N. J., 1962.
8. Wilkinson, W. L., "An Algorithm for Universal Maximal Dynamic Flows in a Network," *Opns. Res.*, Vol. 19, pp. 1602~1612, 1971.