

대규모 시간전개형 네트워크에서의 효율적 최대유량 해법 - Efficient Maximal Flow Algorithms in a Large Time-Expanded Network -

이 달 상*
Lee, Dal-Sang

Abstract

We consider the problem of scheduling a maximal amount of additional, low priority transport through a large multiperiod network, given that we may not interfere with an existing schedule for high priority transport. The problem is transformed into the Time-Expanded network(TENET) without traverse time using TENET Generator (TENETGEN). We describe two specialized heuristic algorithms that guarantee the optimal solutions and show the effectiveness of them by comparing quite favorably with Dinic.

1. 서론

출발지 s 와 목적지 d 사이에 단 하나의 경로 (path)만이 존재한다고 할 때 출발지에 목적지로 가는 여러 종류의 유도체들이 있으며 이러한 여러 종류의 유도체들이 경로상의 각 호(arc)를 통과하는 시간은 각각 다를 수 있다고 하자. 또한 유도체들의 우선순위가 정해져 있어서 통과우선순위가 높은 유도체가 한 호를 통과할 시에는 통과우선순위가 낮은 유도체는 통과우선순위가 높은 유도체가 통과할 때까지 마디(node)에 머물러 있어야 하며 이를 위해 각 마디에는 유도체가 머물수 있는 장소가 있으며 용량(한번에 머물 수 있는 유도체들의 최대 수)이 부과되어 있다. 또 각 유도체 사이에는 안전을 위해 일정한 시간 간격이 유지되어야 한다. 한편 다른 두마디간의 경로들이 출발지 s 와 목적지 d 사이의 본 경로의 일부를 사용할 수도 있으며 (Fig.1 참조), 이때는 본 경로보다 통과우선순위가 높다고 가정한다.

본 연구의 목적은 통과우선순위가 높은 유도체의 경로 통과 일정이 알려져 있을 때, 현재 통과 우선순위가 높은 유도체의 유도체 시각표를 변경시키지 않으면서 일정 기간 동안 보낼 수 있는 통과우선순위가 낮은 유도체의 최대흐름량과 그때의 일정을 결정하는 방법을 구하고자 한다. 이러한 문제는 동적 네트워크(Dynamic Network)의 최대 동적 유량(Maximum Dynamic Flow)문제의 변형으로 지체호를 사용하는 시간 전개형 네트워크로 변환시켜 해를 구할 수 있다. 적용되는 분야로서는 차량,항공기,열차의 스케줄 문제[Orlin,1982]뿐만 아니라 PCB mounter에 의한 PCB 기판조립시간 계산에도 사용될 수 있다.

본 연구에서는 위 문제를 Ford-Fulkerson 모형[1958,1962]을 이용하여 이동시간이 제거된 시간 전개형 네트워크(Time-Expanded NETwork ; TENET)로 변환하고 이달상(1991)의

* 동의대학교 산업공학과

TENET Generator를 사용하여 TENET의 데이터를 구한다. 또한 변환된 TENET의 구조적 단순성과 방향성을 이용하여 해를 찾는 간단한 휴어리스틱 알고리즘들을 제시하고 대규모 네트워크에서 이 휴어리스틱 알고리즘과 Dinic의 최대 흐름 해법과의 비교를 통하여 기존의 해법보다 우수함을 보인다.

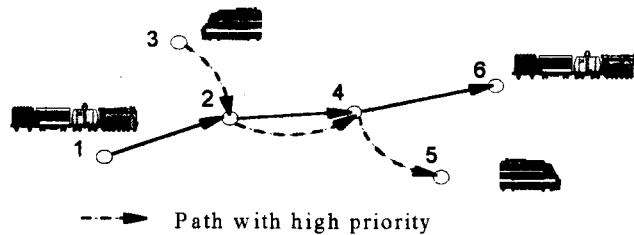


Fig. 1 A path with a high priority

2. 시간 전개형 네트워크에 관한 기존의 연구

동적 네트워크(Dynamic Network)를 시간에 걸쳐 확장함으로 유량을 표현하는 시간 전개형 네트워크(Multiperiod Network, 혹은 Time Expanded Network)에 대한 연구들 중 이론적으로 중요한 몇 가지를 보면 Ford and Fulkerson(1958)은 T기간동안 한 마디에서 다른 마디로 보낼 수 있는 최대유량과 이 최대유량을 얻기위한 스케줄을 구하기 위한 효율적인 알고리즘을 개발하였다. 이 알고리즘의 특성은 T기간에 대해 풀 때 T기간 보다 작은 모든 작은 시간에 대한 최적해가 부산물로 얻어지고 주어진 T기간 동안에 대한 최적해는 시간반복유량이라는 비교적 적은 수의 경로유량(chain-flow)으로 표시될 수 있음과 중간 마디에서 지체가 필요 없음을 보인다. Wilkinson(1971)은 Ford-Fulkerson(1958)해법을 수정해서 공통적 최대 동적 유량(universal maximal dynamic flow)이라 명명한 운송일정을 구하였고 시간반복유량과 공통적 유량과의 차이를 예시하였다. 여기서 공통적 유량이라 함은 P기간동안의 일정을 구축함에 있어 P보다 작은 기간에서 끝날 때 같은 일정이 최대인 유량을 말한다. Minieka(1973)는 다음의 2가지 성질을 증명하였는데 첫째 만약 출발지들에서 한 주어진 출발 형태를 갖는 최대유량과 도착점들에서 도착 형태를 갖는 최대유량이 존재한다면 그때 출발지들과 도착점들에서 이 출발 형태와 도착 형태를 모두 갖는 유량이 존재하고 둘째 출발지들에서의 최지(latest)[최조(earliest)] 출발일정과 도착점들에서의 최조[최지]도착일정을 동시에 갖는 최대 동적유량이 존재함을 보였다. Bellmore and Vemuganti(1973)는 Ford-Fulkerson(1958)의 단일 품목에 대한 최대 동적유량은 최대 시간반복유량으로 나타낼 수 있음이 다품목 최대 동적유량문제에 대해서는 더 이상 적용될 수 없음을 보였다. 대신 충분히 큰 기간에 대해 다품목 최대 동적유량과 시간적 반복 다품목 유량과의 차이가 상수에 의해 한정(bound)되어짐을 보였다. Orlin(1983)은 기존 연구중심 사상이 네트워크를 통과하는 유량인데 반하여 Orlin은 네트워크를 순환하는 유량(throughput)에 초점을 맞추었다. 이 유량을 최대화 하는 문제에서 이 유량이 매 기간 일정함을 보이고 무한 기간의 최대 동적유량 문제의 특수한 경우로 나타낼 수 있음을 증명하였다. 또한 Orlin(1982)은 최대 동적 네트워크 유량문제의 특수한 경우로서 고정된 정기 일정을 만족하는 차량의 수를 최소화하는 문제를 다루었다.

이달상 등(1990,1991)은 단일 경로에서 유도체에 우선순위가 있는 경우의 최대 동적유량문제의 해를 구할 때 기존의 최대 유량해법보다 훨씬 효율적인 열거해법을 구축하였다. 아울러 호의 용량과 이동시간이 변하지않는 경우의 시간 전개형 네트워크발생기를(TENETGEN)을 개발하여 실용성을 향상시켰다.

시간 전개형 네트워크를 응용한 연구들로서 Maxell and Wilson(1981)은 물자 취급 장비(material handling system)들을 시간 전개형 네트워크를 사용한 몇가지 micro-model로 나타내고 이 micro-model들을 결합하여 물자 취급 시스템(material handling system)을 구성하였다. Kroll(1989)은 Maxwell and Wilson(1981)에 의해 제안된 동적 네트워크 모델을 확장시키고 수정해서 유연생산시스템(Flexible Manufacturing System)의 일정계획문제에 대한 최적해를 구하는데 사용하고 계산적으로 가능성을 보였다. Graves and Orlin(1985)은 확정적 수요와 주기적인 데이터를 가진 무한 기간의 로트크기 결정문제를 동적 네트워크를 사용하여 해를 구하였는데 목표는 기간당 평균 비용과 할인 비용을 최소화하는 것이었다.

3. 최대 동적 유량 문제와 시간 전개형 네트워크

동적네트워크(dynamic network : DNET)란 호의 용량 및 비용 파라메타 그리고호를 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 P 동안의 최대 동적 유량을 구하는 문제이다. 이를 정식화하는 과정은 다음과 같다.

마디집합 N인 출발지 s와 목적지 d를 가진 유방향(directed) 네트워크 G(N,A)가 주어졌을 때 비음정수 c(x,y;τ)와 a(x,y;τ)를 각각 호집합 A에서 각 호(x,y)에 대해 시간 τ에서의 용량과 이동시간이라 하자. g(x,y;τ)를 시각 τ에서 (x,y)를 따라 x를 떠나는 유량이라 하면 결과적으로 시각 τ + a(x,y;τ)에 y에 도착한다. 또한 g(x,x;τ)는 마디 x에서 시각 τ 부터 τ+1까지 지체를 나타낸다. 만약 V(P)가 0 에서 1, 1에서 2, ..., P-1 에서 P까지, P기간 동안 s를 떠나거나 d로 들어오는 순유량(the net flow)라 하면 문제는 다음과 같은 선형계획 문제로 표현될 수 있다.

$$\text{Maximize } V(P) \tag{1}$$

subject to

$$\sum_{\tau=0}^{P-1} \sum_{y \in N} \{g(x,y;\tau) - g[y,x;\tau - a(y,x;\tau)]\} - V(P) = 0, \tag{2a}$$

$$\sum_{y \in N} \{g(x,y;\tau) - g[y,x;\tau - a(y,x;\tau)]\} = 0, \tag{2b}$$

(x ≠ s, d; τ = 0, 1, ..., P)

$$\sum_{\tau=0}^{P-1} \sum_{y \in N} \{g(d,y;\tau) - g[y,d;\tau - a(y,d;\tau)]\} + V(P) = 0, \tag{2c}$$

$$0 \leq g(x,y;\tau) \leq c(x,y;\tau) \tag{2d}$$

여기서 node x에서의 지체를 위해 a(x,x;τ) = 1, c(x,x;τ) = ∞

만약 g(x,y;τ)와 V(P)가 (2)를 만족하면 g를 P기간동안 값 V(P)를 가진 s에서 d까지의 동적유량이라 한다. 또한 만약 V(P)가 최대이면, 그때 g는 최대 동적유량이다.

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

그러나 위의 이동 시간이 제거된 시간 전개형 네트워크에서는 지체를 피하는 최대 동적 유량이 항상 존재하기 때문에 이 지체호에 부과된 용량은 중요하지 않았다. 그러나 통과 우선순위가 높은 유도체에 의해 기존의 시간 전개형 네트워크가 수정되면 지체호를 사용해야 하는 최대 동적 유량이 존재하게 되며 Fig. 1에서 나타날 수 있는 한 예가 Fig. 2에 나타나 있다. Fig. 2에서 최대 유량을 구하려는 경로는 1 → 2 → 4 → 6이고 기존 유도체가 각 마디 사이를 이동하는 시간은 1시간이며 마디 2와 마디 4는 경로 3 → 2 → 4 → 5의 일부로 쓰여질 수 있다. 또한 통과 우선 순위가 높은 유도체가 시각 4 부터 마디 1과 마디 6사이의 경로를 사용함을 나타낸다. 여기서 지체호를 사용하지 않는 경로는 1개, 지체호를 사용하는 경로가 2개로 총 3개의 경로를 갖는다.

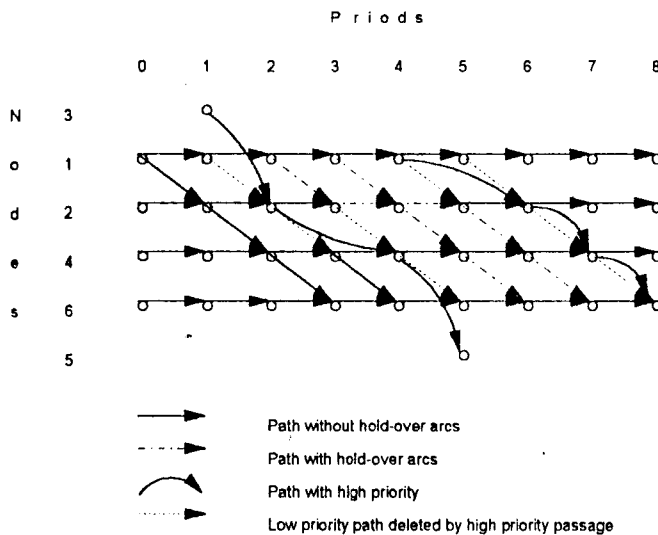


Fig. 2 Maximal dynamic flow using the hold-over arcs

4. 단일 경로의 최대 흐름 스케줄링을 위한 TENET 모형

4.1 기호 설명

P : 지정된 시간

b : 각 유도체가 유지해야 하는 최소한의 시간 간격

N_{sd} : 출발지 s 와 도착지 d 사이의 마디의 수

n_i : 마디 i 가 대피시킬수 있는 통과 우선순위가 낮은 유도체의 수
($i=1,2,3,\dots,N_{sd}+2$)

$t_i^{(k)}$: 유도체 k 의 마디 i 와 마디 $i+1$ 사이의 이동시간

$k=1$: 통과 우선순위가 높은 유도체

$k=2$: 통과 우선순위가 낮은 유도체

4.2 가정

- 1) 출발지 s와 목적지 d는 한개의 경로로 연결 되어 있다.
- 2) 각 유도체의 마디에서 출발과 도착시간 간격은 안전을 위해 b시간 이상 이어야 한다.
- 3) 유도체 k의 지점 i와 지점 i+1 까지 이동시간 t_i 는 b의 정수배이다.
- 4) 각 유도체가 출발지 s를 떠나는 시간 역시 b의 정수배이다.
- 5) 통과 우선 순위가 높은 유도체가 한 지점을 통과할 때는 그 지점을 통과 하는 우선순위가 낮은 유도체는 대피하여 그 유도체가 통과할 때까지 기다려야 한다.
- 6) 각 마디는 용량을 갖는다.

출발지는 마디 1로 도착지를 마디 $N_{sd}+2$ 로 하면 마디(i,j)는 시각 j에서의 마디 $i(i=1,2,\dots,N_{sd},N_{sd}+1,N_{sd}+2)$ 그리고 $j = 0,b,2b,\dots,P/b$ 를 나타낸다. 특히 마디(1,0)은 출발지, 마디 $(N_{sd}+2,P/b)$ 는 목적지를 의미한다.

오직 한 유도체(K=2)가 시각 j에서 마디 i를 떠나서 마디 i+1에 도착하기 전 K=1인 유도체에 의해 추월되지 않을 경우에만 마디(i,j)와 마디(i+1,j+t_i)에 용량이 1인 호 ((i,j),(i+1,j+t_i))가 존재하게 된다. 또한 $j=0,b,2b,\dots,P/b$ 에 대해 마디(i,j)와 마디(i,j+1) 사이에 용량이 n_i 인 호 ((i,j),(i,j+1))이 존재하며 이외의 다른 호는 존재하지 않는다. 예를 들어 k=1인 유도체가 마디 i를 출발하여 마디 i+1에 닿기 전에 이동시간 t_i 를 가진 k=2인 유도체를 추월할 수 있다면 그림 3.3은 TENET의 일부분을 표현하게 된다. 이러한 구조를 수리 모형으로 작성하면 목적함수는 마디(1,0)에서 마디($N_{sd}+2,P/b$)까지의 최대 흐름량을 구하는 것이 된다.

4.3 수리모형과 시간 전개형 네트워크 발생기(TENETGEN)

$G_T = (N^P, A^P)$ 에서 통과우선순위가 높은 유도체에 의해 호가 삭제되고 지체호가 추가됨으로 수정된 유한 유방향 그래프를 $G_T(X, \Gamma)$ 라 하자.

여기서 X는 수정된 마디의 집합이고 Γ 는 수정된 호의 집합이다. 우선 구하려는 최대유량에 대한 식 구성은

$$\begin{aligned} & \text{Maximize } V & (3) \\ & \text{Subject to} \end{aligned}$$

$$\sum_{\langle u,v \rangle \in X} f\langle l,m \rangle \langle u,v \rangle - \sum_{\langle w,z \rangle \in X} f\langle w,z \rangle \langle l,m \rangle = \begin{cases} V & \langle l,m \rangle = s \\ 0 & \langle l,m \rangle \neq s, d, \\ -V & \langle l,m \rangle = d \end{cases} \quad \text{for all } \langle l,m \rangle \in X \quad (3a)$$

각 호 $\langle l,m \rangle, \langle u,v \rangle \in \Gamma$ 에 대해

$$0 \leq f\langle l,m \rangle \langle u,v \rangle \leq C\langle l,m \rangle \langle u,v \rangle \quad (3b)$$

여기서

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

$$l, u, v \in I = \{I: I=1, 2, \dots, N_{sd}+2\}$$

$$m, v, z \in J = \{J: J=0, b, 2b, \dots, P/b\}$$

X : 수정된 그래프 G_T' 의 모든 마디(node)의 집합

Γ : 수정된 그래프 G_T' 의 모든 호(arc)의 집합

V : 구하고자 하는 총 유량

f_{ij} : 호(i,j)의 흐름량

C_{ij} : 호(i,j)의 용량

DNET G를 TENET G_T 로 변환 할때 지정된 시간 P가 커지면 변환 네트워크 G_T 는 구조는 복잡하지 않으나 크기가 커지는 단점이 있다. 예를들어 출발지와 도착지를 포함하여 통과 지점이 100개이고 지정기간 P가 24시간, 시간간격이 3분일 경우 총 마디 수가 48000개 호 수는 대략 96000개인 네트워크로 된다. 이 처럼 큰 네트워크를 수작업으로 입력시킨다는 것은 매우 불편한 일이며, 따라서 데이터를 간단히 입력시킬 수 있는 방법을 구하게 된다. 이에 이달상(1990,1991)은 TENET의 호방향의 규칙성을 이용하여 TENETGEN이라고 명명한 TENET 발생기(Generator)를 개발하였다.

TENETGEN의 기본적인 개념은 각 마디사이에 통과 우선순위가 낮은 유도체의 진행 호를 발생시켜 놓고, 통과 우선순위가 높은 유도체의 진행 호가 발생될때 TENET 구조의 규칙성을 따라 이미 생성된 통과 우선순위가 낮은 유도체의 진행 호를 소거해가는 것이다. 여기에서 생성된 시간전개형 네트워크(NENET)는 각 마디에서 나오고 들어가는 호는 각각 최대 2개 이고 목적지으로 향하는 단방향의 특성을 갖고 있다.

5. 알고리즘

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

I. TENETGEN을 이용한 TENET의 발생

- 단계 1 : 중간마디의 수(N_{sd}),계획기간(P),시간간격(b),
- 단계 2 : 각 마디에서의 용량(n_i)를 입력 ($i=1,2,\dots,N_{sd}+2$)
- 단계 3 : 통과 우선순위가 높은 유도체과 통과 우선순위가 낮은 유도체의 각 마디간 운행시간 $T^{(1)}, T^{(2)}$ 를 입력($i=1,2,\dots,N_{sd}+1$)
- 단계 4 : 통과 우선순위가 높은 유도체의 출발지에서의 발차시각 (PST)을 입력
- 단계 5 : TENETGEN을 실행

II. 최대 흐름 스케줄링 결정

I의 단계 5에서 발생된 DATA로 다음의 휴어리스틱 알고리즘들 중의 하나를 선택하여 해를 구함

1. 알고리즘 1 (E1) - 깊이 탐색(Depth-first search)를 사용

단계 1

- (1.1) 출발지에서 우선 순위가 낮은 유도체의 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, $j+t^{(2)}$)로 가는 호가 있는지를 조사한다.
 - (2.2.1) 만약 있으면 마디 (i,j)에 표지하고 마디 (i,j)로 부터 마디 (i+1, $j+t^{(2)}$)로 가는 호의 용량을 1 만큼 감소시키고 $i=i+1, j=j+t^{(2)}$ 로 놓고 (2.1)로 간다.
 - (2.2.2) 없으면 마디 (i,j)에 표지하고 마디 (i,j)로 부터 마디 (i,j+d)로 가는 호가 있는지를 조사한다.

- (2.2.2.1) 있으면 마디 (i,j)로 부터 마디 (i,j+d)로 가는 호의 용량을 1 만큼 감소시키고 $j=j+d$ 로 놓고 (2.1)로 간다.
- (2.2.2.2) 없으면 마디 (i-1,j)로 부터 마디 (i,j)까지 경로상의 호들의 용량을 1씩 증가시키고 $i=i-1$ 로 놓고 (2.2.2)로 간다.

단계 3

- (3.1) 경로의 수를 1 만큼 증가시키고 지금까지의 표지를 모두 지우고 단계 1로 간다.

2. 알고리즘 2 (E2)

먼저 단계 1, 단계 2중 (2,1)(2,2),(2,2,1)과 단계 3으로 구성된 해법절차를 통해 부분해를 구한 다음 알고리즘 1(E1)을 다시 적용하여 최적해를 구한다.

6. 실험 결과 및 비교

6.1 계산량(complexity)의 비교

생성된 시간 전개형 네트워크(TENET)에서 구조적 단순성과 단방향성을 이용하여 경로들을 살펴 보면 출발지에서 목적지로 가는 최대 경로의 수는 P/b 개이다. 이것은 각 마디간 cut의 최대 크기를 의미한다. 또한 각 경로는 최소 $N_{sd}+1$ 개, 최대 $(N_{sd}+1+P/b)$ 개의 호로 구성된다. 휴어리스트릭 알고리즘들은 최악의 경우 각 마디에서 $N_{sd} * (N_{sd}+P/b)$ 만큼 계산하게 되나 실제로는 마디가 진행될 때마다 계산량은 급격히 감소된다. 이론적인 계산량은 $O(N^3)$ 로 표시할 수 있다.

이것은 Dinic 해법의 계산 복잡도, $O(N^2M)$, Karzanov해법의 계산 복잡도, $O(N^3)$, MPM 해법의 계산 복잡도, $O(N^3)$ [Goldfarb and Grigoriadis,1988; Nemhauser and Kan,1989]와 비슷하나 실제로는 휴어리스트릭 알고리즘들이 해를 구하는 계산량이 더 적음을 보장하는데, 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 워크스테이션 RS/6000을 사용하였다. 계산시간은 입출력에 소요되는 시간은 배제한 시스템 시간을 측정하였다. 실험을 시행한 결과의 일부가 Fig. 3과 Table 1에 나타나 있다.

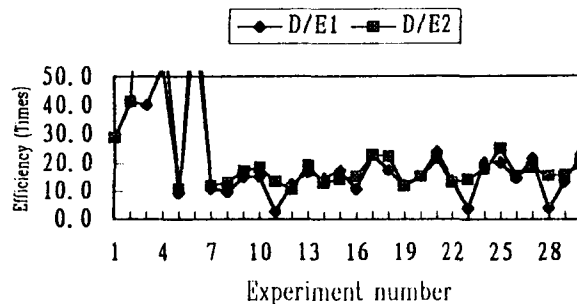


Fig. 3 The efficiency of algorithms

Table 1. Results of Experiments

Execution time : Second

No	Ncd	b	N	M	HPP	NPO	E1	E2	D	D/E1	D/E2
1	10	0.06	4812	8778	19	328	0.02	0.02	0.58	29.0	29.0
2	10	0.05	5772	10561	22	402	0.02	0.02	0.83	41.5	41.5
3	10	0.04	7212	13203	28	507	0.03	0.01	1.21	40.31	21.0
4	10	0.03	9612	17692	37	691	0.04	0.03	2.14	53.5	71.3
5	30	0.06	12832	24026	19	258	0.08	0.07	0.75	9.4	10.7
6	10	0.02	14412	26860	55	1088	0.07	0.06	4.83	69.0	80.5
7	30	0.05	15392	28911	22	331	0.10	0.09	1.08	10.8	12.0
8	30	0.04	19232	36112	28	428	0.16	0.12	1.57	9.8	13.1
9	50	0.06	20852	39469	19	220	0.08	0.07	1.21	15.1	17.3
10	50	0.05	25012	47467	22	290	0.11	0.09	1.67	15.2	18.6
11	30	0.03	25632	48190	37	596	0.97	0.20	2.67	2.8	13.4
12	70	0.06	28872	54700	19	164	0.12	0.14	1.49	12.4	10.6
13	50	0.04	31252	59317	28	397	0.15	0.13	2.49	16.6	19.2
14	70	0.05	34632	65795	22	238	0.15	0.17	2.20	14.7	12.9
15	90	0.06	36892	70249	19	116	0.09	0.11	1.56	17.3	14.2
16	30	0.02	38432	72184	55	933	0.53	0.37	5.66	10.7	15.3
17	100	0.06	40902	77664	19	80	0.07	0.07	1.60	22.9	22.9
18	50	0.03	41652	79040	37	564	0.23	0.18	4.00	17.4	22.2
19	70	0.04	43272	82234	28	345	0.24	0.25	2.95	12.3	11.8
20	90	0.05	44252	84420	22	180	0.14	0.14	2.13	15.2	15.2
21	100	0.05	49062	93315	22	150	0.09	0.10	2.15	23.9	21.5
22	90	0.04	55292	105551	28	289	0.22	0.23	3.02	13.7	13.1
23	70	0.03	57672	109549	37	516	1.18	0.33	4.65	3.9	14.1
24	100	0.04	61302	116707	28	253	0.16	0.18	3.21	20.1	17.8
25	50	0.02	62452	118366	55	893	0.41	0.33	8.27	20.2	25.1
26	90	0.03	73692	140674	37	464	0.36	0.34	5.26	14.6	15.5
27	100	0.03	81702	155531	37	428	0.23	0.27	4.95	21.5	18.3
28	70	0.02	86472	163935	55	848	2.30	0.59	9.12	4.0	15.5
29	90	0.02	110492	210466	55	796	0.72	0.60	9.49	13.2	15.8
30	100	0.02	122502	232620	55	760	0.42	0.49	9.55	22.7	19.5

Time horizon : 24 hours

Nsd : Number of intermediate nodes

N : Number of nodes

b : Time interval between passages(hour)

M : Number of arcs

HPP : No. of occurrences in high priority passage

NPO : No. of paths

E1 : Enumeration Algorithm 1

E2 : Enumeration Algorithm 2

D : Dinic's Algorithm

위 실험에서는 본 연구에서 개발한 휴어리스틱 알고리즘들과 Dinic의 최대 유량 해법을 비교, 검토하기 위하여 IBM 워크스테이션 RS/6000에서 실행할 때 다룰 수 있는 최대 크기의 네트워크를 주로 다루었으며, 각 마디에서 유도체를 대피시킬 수 있는 장소의 크기는 유도체의 3-5배 사이에서, 각 유도체의 각 마디간 이동시간은 b의 3-4배 내에서, 우선 순위가 높은 유도체의 발생 수는 P/b 의 4.5%에서 난수를 발생시켜 임의로 주었다.

Table 1에서 보이듯이 휴어리스틱 알고리즘이 Dinic 해법보다 10-20배 정도 해를 구하는 속도가 빨랐다. 위 실험 결과 중 최대 실행 시간은 Dinic 해법에서 9.55초인 반면 휴어리스틱 알고리즘은 2.30초였으며 대부분 1초를 넘지 않았다.

7. 결 론

본 연구에서는 경로의 최대 흐름 일정을 결정하는데 이달상(1990, 1991)이 개발한 시간 전개형 네트워크(TENET)모형과 TENET 발생기(TENETGEN)사용하여 여기에서 얻어진 네트워크의 구조적 단순성과 방향성을 이용하여, 최적해를 구하는데 깊이 탐색만을 이용한 것과 지체호를 사용하지 않는 단순 경로를 먼저 구한 후 깊이 탐색을 적용한 휴어리스틱 알고리즘들을 개발하였다. 이 휴어리스틱 알고리즘들과 Dinic의 최대 유량 해법의 계산량(complexity)을 비교 검토하고, 대규모의 문제에 대한 해결 능력을 알아보기 위해 IBM 워크스테이션 RS/6000에서 Dinic의 최대유량해법이 다룰 수 있는 최대 크기(마디:122,502개, 호:232,620개)까지 실험을 해 본 결과, 휴어리스틱 알고리즘들이 Dinic의 해법보다 평균 15배 정도 빠른 결과치를 얻었다. 이것은 Dinic 해법이 네트워크의 구조와는 무관하게 최대 유량을 구하기 위해 층(layer) 네트워크를 구성하는 바, 이에 소비되는 시간이 많았기 때문이라 생각된다. 그리고 지체호를 사용하지 않는 단순 경로를 먼저 구한 후 깊이 탐색을 적용한 휴어리스틱 알고리즘들이 깊이 탐색만을 사용한 방법보다 해를 구하는 속도의 편차가 적어 더 우수한 것으로 나타났다.

또한 본 휴어리스틱 알고리즘들이 차지하는 기억용량은 Dinic 해법보다 50% 이상 작아서 실제로는 훨씬 더 큰 규모의 문제를 다룰 수 있다. 본 연구에서 구한 휴어리스틱 알고리즘들을 IBM 워크스테이션 RS/6000에서 행한 실험에서는 약 50만개의 마디와 약 100만개의 호를 동시에 다룰 수 있었다.

참 고 문 헌

1. Bellmore, M. and R.R. Vemuganti, 1973; "On Multi-Commodity Maximal Dynamic Flows," *Operations Research*, Vol.21, No.1, pp. 10-21.
2. Ford, L.R. and D.R. Fulkerson, 1958; "Constructing Maximal Dynamic Flow from Static Flows," *Operations Research*, Vol.6, pp. 419-433.
3. Ford, L.R., Jr. and D.R. Fulkerson, 1962; *Flows in Networks*, Princeton University Press, Princeton.
4. Goldfarb, D. and M.D. Grigoriadis, 1988; "A Computational Comparison of the Dinic and Network Simplex Methods for Maximum Flow," *Annals of Operations Research*, Vol.13, pp. 83-123.
5. Graves, S. C. and J. B. Orlin, 1985; "A Minimum Concave-Cost Dynamic Network Flow Problem with an Application to Lot-Sizing," *Networks*, Vol.15, pp. 59-71.
6. Jensen, P. A. and J. W. Barnes, 1980; *Network Flow Programming*, John Wiley & Sons, New York.
7. Kroll, D. E., 1989; *The Use of Dynamic Network in Scheduling Flexible Manufacturing Systems*, Ph D. Dissertation, University of Illinois at Urbana-Champaign.
8. Maxwell, W.L. and R.C. Wilson, 1981; "Dynamic Network Flow Modeling of Fixed Path Material Handling Systems," *AIIE Transactions*, Vol. 13, No. 1, pp. 12-21.
9. Minieka, E., 1973; "Maximal, Lexicographic, and Dynamic Network Flows," *Operations Research*, Vol. 21, No. 2, pp. 517-527.

10. Nemhauser, G. L. and A. H. G. Rinnooy Kan, 1989; *Handbooks in Operation Research and Management Science : Optimization*, Vol. 1. Chap. IV, North-Holland.
11. Orlin, J. B., 1981; *Dynamic Network Flows*, Ph.D. Dissertation, Stanford University.
12. Orlin, J. B., 1982; "Minimizing the Number of Vehicles to Meet a Fixed Periodic Schedule : An Application of Periodic Posets," *Operations Research*, pp. 760-776.
12. Orlin, J. B., 1983; "Maximum - Throughput Dynamic Network Flows," *Mathematical Programming*, Vol. 27, pp. 214-231.
13. Wilkinson, W. L., 1971; " An Algorithm for Universal Maximal Dynamic Flows in a Network," *Operations Research*, Vol. 19, pp. 1602-1612.
14. 이 달상, 김 만식, 1990; "시간 전개형 네트워크를 이용한 선로의 최대흐름 스케줄링," 대한교통학회지, 제8권 2호, pp. 67-75.
15. 이 달상, 김 만식, 이 영혜, 1991; "지체호를 사용하는 시간 전개형 네트워크의 개발," 산업공학, 제4권 2호, pp. 25-34.