

■ 論 文 ■

# Conjugate Gradient 법을 이용한 경로기반 통행배정 알고리즘의 구축

A Development of a Path-Based Traffic Assignment Algorithm  
using Conjugate Gradient Method

**강 승 모**

(서울대학교 도시공학과 연구원)

**권 용 석**

(서울대학교 도시공학과 연구원)

**박 창 호**

(서울대학교 지구환경  
시스템공학부 교수)

## 목 차

- I. 서론
- II. 경로기반통행배정
  - 1. 경로기반통행배정의 의의
  - 2. Gradient Projection 알고리즘
- III. 통행배정 알고리즘의 구축
  - 1. 알고리즘의 이론적 배경
- 2. Conjugate Gradient 반복회수
- 3. 통행배정알고리즘의 단계
- IV. 알고리즘 적용
- V. 결론 및 향후 연구과제
- 참고문헌

Key Words : Conjugate Gradient 법, Gradient Projection 알고리즘, 경로기반통행배정, 교통 최적화 기법

## 요 약

경로기반 통행 배정은 실시간 통행 배정에서 이용되는 경로기반 해를 제공할 수 있기 때문에 첨단 교통 체계 (ITS)의 실시간 교통 제어 및 교통 안내 등에 유용하게 이용될 수 있다.

많이 사용되고 있는 경로기반 통행배정 알고리즘의 하나인 Gradient Projection(GP) 알고리즘은 일반적으로 최적해 근처로는 빠른 접근 속도를 보이나, 일단 최적해에 근접하면 수렴 속도가 다소 느려지게되는 단점이 있다.

기존 알고리즘의 이러한 단점을 극복하기 위해 기존의 GP 알고리즘에 Conjugate Gradient 법을 결합시켜 보다 효율적인 경로기반 통행배정 알고리즘을 구축하였다. 이는 최적해 근처에서 더욱 정확한 이동방향을 결정 하여 빠른 시간 내에 최적해를 도출해 내도록 하기 위한 것이다.

또한, 구축된 알고리즘을 가로망에 적용, 그 효율성을 검증하여 Conjugate Gradient 법이 통행 배정 모형의 사용자 평형 모형에서와 같은 목적함수의 경우에서도 매우 빠른 수렴을 위해 유용하게 쓰일 수 있다는 것을 보였다.

## 1. 서론

교통 체계에 전자, 통신, 제어, 정보, 컴퓨터 등의 부대기술을 응용한 첨단 교통 체계(ITS)는 교통 혼잡을 경감시키고 교통 안전을 증진시키며, 여행자 정보를 확장시킴으로써 기존 도로망의 운영을 혁신적으로 개선할 목적으로 개발되고 있다. 이러한 첨단 교통 체계(ITS)의 하위 구조로서 첨단 여행자 정보체계(AVIS)나 첨단 교통관리(ATMS) 등에서는 실시간 기종점 수요의 추정과 동적 배정 체계가 필요하고, 이를 위해서는 실시간 통행배정이 필수적이다.

이 같은 동적 통행 배정의 실시간 적용에는 보통 정적 통행배정의 결과로서 경로기반 해를 필요로 하게 된다.<sup>1)</sup> 이는 링크 단위의 해만을 제공하는 기존의 전통적인 통행배정 알고리즘에서는 구할 수 없는 것이다. 즉, 지금까지 일반적으로 사용된 Frank-Wolfe(F-W) 알고리즘과 같은 링크기반 통행배정(LBA: Link-Based Assignment) 알고리즘들은 이용자의 전체 통행경로를 파악할 수 있는 경로 패턴을 도출하기 힘들기 때문에, 수요 예측결과를 이용한 교통 운영 및 교통관리 등을 위한 다양한 적용에서 제한적일 수밖에 없다.

최근 이러한 기존 통행배정 방법들의 문제점을 해결하기 위해 경로기반 통행배정 (PBA: Path-Based Assignment)이라고 불리는 새로운 통행배정 방법에 대한 연구가 활발히 이루어지고 있다.

최근에 가장 많이 사용되고 있는 경로기반 통행배정 알고리즘의 하나인 Gradient Projection(GP) 알고리즘은 Newton's Method를 응용한 것으로서 기존의 F-W 알고리즘보다 정확도나 수렴 속도 면에서 훨씬 뛰어난 결과를 보여주고 있다. 하지만 GP 알고리즘은 일반적으로 최적해 근처로는 빠른 접근 속도를 보이나, 일단 최적해에 근접하면 수렴 속도가 다소 느려지게 되는 단점이 있다.

본 연구에서는 GP 알고리즘의 이러한 단점을 극복하기 위해 Conjugate Gradient(CG) 알고리즘을 도입하였다. CG 알고리즘은 타원형의 함수를 최소화시키기 위해 개발된 방법으로 정해진 수의 반복수행 내에서 정확한 해를 찾을 수 있다는 장점이 있다. 일

반적인 비선형 최적화 문제에서의 목적함수들은 최적해 근처에서 타원 함수에 근사될 수 있기 때문에<sup>1)</sup> 통행배정 모형의 사용자 평형 모형에서와 같은 목적함수에서도 매우 빠른 수렴을 위해 유용하게 쓰일 수 있다.

본 연구는 GP 알고리즘에 CG 알고리즘을 결합시켜, 보다 효율적인 경로기반 통행배정 알고리즘을 구축하고, 기존의 통행배정 알고리즘과의 비교를 통해 그 효율성을 검증하였다.

## II. 경로기반 통행배정

### 1. 경로기반 통행배정의 의의

경로기반 통행배정은 기종점간의 경로교통량을 결정변수로 하여 배정문제의 해를 찾는다. PBA는 최근 급격한 컴퓨터 기술의 발달 및 최적화기법의 개발로 대규모 네트워크에서도 사용 가능하게 되었을 뿐 아니라, 기존 통행배정 방법에 비해 월등히 빠른 수렴 속도가 확보되면서 연구가 활발히 진행되고 있다.

일반적으로, 경로기반 배정모형의 해는 링크교통량 측면에서는 유일하지만, 경로교통량 측면에서는 비록 그 해가 최적일지라도 유일하지 않다고 알려져 있다. Sheffi<sup>13)</sup>는 일반적인 배정문제에서 경로교통량이 유일하지 않은 이유를 언급하였고, Jayakrishnan et al.<sup>6)</sup>도 GP알고리즘을 이용한 경로기반 배정모형의 결과가 최적치이지만 유일(unique)하지는 않다고 하였다. 비록 경로 기반의 해가 유일한 것은 아니나 기존의 방법으로는 알기 힘든 경로해를 구할 수 있고, 링크 교통량의 측면에서도 기존의 방법보다 월등히 빠르며, 정확한 해를 구할 수 있으므로, 경로기반 배정모형은 그 자체로서 통행 배정 문제의 해결에 있어 중요한 의미를 가진다고 말할 수 있다.

### 2. Gradient Projection 알고리즘

GP알고리즘은 각각의 반복 계산에서 각각의 기종점교통량 O/D쌍에 대한 최단 경로와 최소 일차미분비용을 기반으로 하여, 반복 수행마다 각 경로에서의 교통량의 변화를 각 경로와 최단경로의 통행비용 차

1) Mahmassani, H. S., and S. Peeta. *Network Performance under System Optimal and user Equilibrium Dynamic Assignment*. Paper presented at 72nd Annual Meeting of the Transportation Research Board, Washington, D. C., January 1993 ( Jayakrishnan et al.<sup>6)</sup>에서 재인용).

이와 목적 함수의 이차미분값 즉, 이차미분비용에 근거하여 계산한다.

이러한 경로 계산 알고리즘은 Steepest Descent Method나 Newton's Method와 같은 제약조건이 없는 일반적인 최적화 방법에 제약조건을 첨가한 기법으로 생각할 수 있다.

GP 알고리즘의 주된 반복 수행 과정은 다음과 같다.

$$f_p^{k+1} = [f_p^k - \alpha \overline{H}_p^{-1} (t_p^k - t_{p_x}^k)]^+ \quad (1)$$

여기서,  $f_p$ 는  $P_w$ 에 속하는 경로  $p$ 의 교통량을 나타내며,  $\alpha$ 는 이동간격,  $\overline{H}_p$ 는 경로  $p$ 의 이차 미분비용,  $t_p^k, t_{p_x}^k$ 는 각각 경로  $p$ 와 최단경로  $\overline{p}_w$ 의 경로비용이다. 이와 같은 과정들을 GP 알고리즘이라고 부르며 GP 알고리즘은  $\alpha^k, \overline{H}_p, (t_p - t_{p_x})$ 들을 이용해 최단경로로 전이되는 양을 결정하는 경로배정 방법이라고 볼 수 있다. 이 과정에서 다음 사항들을 알 수 있다. 첫째, 모든  $p \neq \overline{p}_w$ 에 대해  $t_p \geq t_{p_x}$ 이므로 최단경로가 아닌 곳에 배정되어 있는 양의 교통량은 최소 일차미분비용 경로  $\overline{p}_w$ 에 전이되는 양만큼이 감소된다. 이동간격이 충분히 크다면 모든 양이  $\overline{p}_w$ 로 옮겨갈 것이다. 또한 최단이 아닌 경로의 교통량이 0이라면 계속 0으로 남아있을 것이다. 따라서 식(1)의 반복계산은 양의 교통량을 갖는 경로에 대해서만 수행될 것이다. 마지막으로 반복계산의 시작단계에서는 초기에 양의 교통량을 가지고 있던가, 앞서의 반복계산에서 최소 일차미분비용 경로로 결정된 경로만이 교통량을 가지고 있을 것이다. 이는 계산되는 경로수를 적게 유지하여 계산량을 줄일 수 있게 되므로 상당히 중요하다.

### III. 통행배정 알고리즘의 구축

#### 1. 사용 변수의 정의

본 연구에서 사용된 변수들의 정의는 다음과 같다.

- $W$  : O/D쌍  $w$ 들의 집합
- $P_w$  : O/D쌍  $w$ 간의 사용되는 경로들의 집합
- $\overline{p}_w$  : O/D쌍  $w$ 간의 최단경로
- $S_a$  : 링크  $a$ 의 비용함수
- $\delta_p^a$  : 경로  $p$ 가 링크  $a$ 를 포함하면 1, 아니면 0

- $f_p^k$  :  $k$ 번째 반복수행에서 경로  $p$ 의 교통량
- $t_p^k$  :  $k$ 번째 반복수행에서 경로  $p$ 의 경로비용
- $H_p^k$  :  $k$ 번째 반복수행에서 경로  $p$ 의 이차미분비용
- $r_w$  : O/D쌍  $w$ 간의 교통량
- $\overline{\alpha}$  : 이동간격(step size)
- $d^k$  :  $k$ 번째 반복수행에서 이동방향
- $C^k$  :  $k$ 번째 반복수행에서 CG를 이용해 교통량을 계산할 경로의 집합
- $g^k$  :  $k$ 번째 반복수행에서 일차미분비용
- $x_a$  : 링크  $a$ 의 교통량
- $|v|$  : 벡터  $v$ 에 대한 각 성분의 root mean square 값

### 2. 알고리즘의 이론적 배경

우선, 다음과 같은 사용자 평형 목적함수를 생각한다.

$$\begin{aligned} \text{Minimize } Z(f) &= \sum_{link\ a} \int_0^{\sum_{p \in P_a} \delta_p^a f_p} S_a(x) dx \quad (2) \\ \text{subject to } \sum_{p \in P_x} f_p &= r_w, \quad \forall w \in W \\ f_p &\geq 0, \quad \forall w \in W, p \in P_w, p \neq \overline{p}_w \end{aligned}$$

이를,

$$\begin{aligned} Z(f) &= Z(f_1, f_2, \dots, f_{\overline{p}_w}) \\ &= Z(f_1, f_2, \dots, r_w - \sum_{\substack{p \in P_x \\ p \neq \overline{p}_w}} f_p) = \tilde{Z}(f) \quad (3) \end{aligned}$$

로 정의하면, 제약조건에서 교통량 보존식을 없앨 수 있으며, 다음과 같은 형태의 문제로 표현할 수 있다.

$$\begin{aligned} \text{Minimize } \tilde{Z}(f) \quad (4) \\ \text{subject to } f_p &\geq 0, \quad \forall w \in W, p \in P_w, p \neq \overline{p}_w \end{aligned}$$

이러한 문제에 대해, 각각의 반복수행에서 각 구성 변수들은 다음과 같은 계산과정을 이용해 이동해 간다.

$$f_p^{k+1} = [f_p^k - \overline{\alpha} d_p^k]^+ \quad (5)$$

여기서,  $d^k$ 는 이동방향,  $[\cdot]^+$ 는 투영(projection)을

의미한다.<sup>2)</sup>  $\bar{\alpha}$ 를 결정하는 방법 중에 하나는 이를 상수로 가정하는 것인데 이는 일반적으로 많은 비선형 문제에 있어 매우 효율적이라는 것이 알려져 있다.<sup>3-6,15)</sup>

본 연구에서는 기존 GP 알고리즘에서의 반복수행 과정에 CG 알고리즘을 사용한 하부 구조를 삽입하여 보다 빠르게 최적해를 구할 수 있는 알고리즘을 구축하였다. 즉, 초기에는 기존의 GP에 의한 이동으로 빠르게 최적해 근처로 접근한 후, CG 법을 이용하여 구한 이동간격을 이용하여 적은 반복 수행 내에서 비교적 정확한 해를 찾을 수 있도록 구성된다.

최적해로 접근하기 위한 알고리즘의 k번째 반복수행은 다음과 같이 시작될 수 있다. 우선 집합  $C^k$ 를 다음과 같은 식에 의해 정의한다.  $C^k$ 는 GP로 구한 경로 교통량의 변화율이 어떤 값  $\epsilon$ 보다 작은 경로의 집합으로서 CG를 이용해 교통량을 계산할 경로의 집합이다.

$$C^k = \{p \mid 0 \leq s_p^k \leq \epsilon\} \tag{6}$$

$$s_p^k = \frac{|f_p^k - [f_p^k - \bar{\alpha} \overline{H}_p^{-1}(t_p - t_{p_w})]^+|}{f_p^k} \tag{7}$$

이 결과를 이용하여  $f_p^k$ 은 다음과 같이 나누어진다.

$$f = \begin{bmatrix} \hat{f} \\ \bar{f} \end{bmatrix} \tag{8}$$

여기서  $\hat{f}$ 는  $p \in C_k$ 인 성분  $f_p$ 의 벡터이고  $\bar{f}$ 는  $p \notin C_k$ 인 성분  $f_p$ 의 벡터이다.  $d_p = H_p^{-1}g_p$ 이므로 여기서 이동방향 벡터  $d^k = (\hat{d}^k, \bar{d}^k)$ 는 각각 다음의 식을 풀어 구한다.

$$\widehat{H}^k \hat{d}^k = -g^k \tag{9}$$

$$\overline{H}^k \bar{d}^k = -g^k \tag{10}$$

여기서  $g^k$ 는  $\frac{\partial Z(f)}{\partial f_p}$ 의 벡터로서 식(11)과 같이 계산된다.

$$g^k = t_p^k - t_{p_w}^k \tag{11}$$

여기서,  $t_p^k(t_{p_w}^k)$ 는 경로  $p(\overline{p_w})$ 의 통행시간(또는 통행비용)을 나타낸다. 한편,  $\widehat{H}_p^k$ 와  $\overline{H}_p^k$ 는 모두  $Z(f)$ 의 Hessian 행렬에 근거해서 구하지만,  $\widehat{H}_p^k$ 를 계산에서 Hessian 행렬의 대각항만을 고려하여 구한다면 식(12)와 같게된다. 여기서,  $L_p$ 는 경로  $p$ 와 최단 경로  $\overline{p_w}$ 중에 한 경로에만 속하는 링크의 집합이다.

$$\overline{H}_p^k = \sum_{\substack{\text{link } a \\ \in L_p}} \frac{\partial S_a^k}{\partial x_a^k} \tag{12}$$

이를 이용해 결국  $p \in C_k$ 인 경로  $p$ 에 대한 반복수행 과정을 구하면 식(13)과 같고, 이는 GP 알고리즘에서의 반복수행 과정과 동일하다.<sup>6)</sup>

$$f_p^{k+1} = [f_p^k - \alpha \overline{H}_p^{-1}(t_p^k - t_{p_w}^k)]^+, p \in C^k \tag{13}$$

### 3. Conjugate Gradient 반복수행

여기서는 CG 알고리즘을 이용하여 식(9)의  $\hat{d}^k$ 를 구하는 과정을 보일 것이다. 이 과정은 알고리즘 상에서  $p \in C_k$ 인 경로  $p$ 에 대한 경로 교통량의 갱신을 위한 CG 반복수행을 나타낸다.  $z$ 를  $\hat{d}^k$ 의 근사해라고 가정하면, 다음과 같은 제약없는 최소화 문제를 통해  $z$ 를 구할 수 있다.

$$\min_z g^k \cdot z + \frac{1}{2} z' \widehat{H}^k z \tag{14}$$

위 식에서  $\cdot$ 는 전치행렬을 나타낸다.

positive definite이고 대칭인 행렬  $P_k$ 가 선택된다면, 이동간격  $\hat{d}^k$ 을 나타내는  $\{z_m\}$ 은 다음과 같은 반복수행을 통해 구할 수 있다.

$$z_0 = 0 \\ z_{m+1} = z_m + \gamma_m u_m, m = 0, 1, \dots, \tag{15}$$

이 때, Conjugate 방향 간격  $\{u_m\}$ 은 다음과 같은

2)  $[x]^+ = \max\{0, x\}$

순환적인 방법에 의해 구해진다.

$$\begin{aligned} u_0 &= -\overline{H}_k r_0, \\ u_m &= -\overline{H}_k r_m + \beta_m u_{m-1}, \quad m=1,2,\dots \end{aligned} \quad (16)$$

오차  $\{r_m\}$ 은 다음과 같이 주어지고

$$r_m = \widehat{H}^k z_m + g_k \quad m=0,1,\dots, \quad (17)$$

$\gamma_m$ 과  $\beta_m$ 은 scalar로서 다음과 같다.

$$\gamma_m = \frac{r_m' \overline{H}_k r_m}{u_m' \widehat{H}^k u_m} \quad m=0,1,\dots \quad (18)$$

$$\beta_m = \frac{r_{m-1}' \overline{H}_k r_m}{r_{m-1}' \overline{H}_k r_{m-1}} \quad m=0,1,\dots \quad (19)$$

Luenberger<sup>10)</sup>에 의하면 이 방법은  $\overline{H}_k$ 에 관계없이 식(9)의 해  $\widehat{d}^k$ 를 적어도 n-1 번째(n=계산되는 경로의 개수)에서 찾게된다. 즉,  $\widehat{d}^k = z_{n-1}$ 이다. 그러나 이 연구에서는 단지 적은 수의 반복 계산이 수행되는 상황에서 알고리즘의 효율성을 최대한 높일 수 있는 근사해를 도출하는데 관심이 있기 때문에 위의 문제를 정확하게 푸는 것보다 적당한 단계까지 만의 결과를 이용하는 것으로 충분하다. 즉, 적당한 기준에서 반복횟수가 정해진 수를 넘거나 어떤 값  $\phi^k$ 에 대해  $|r_m| \leq \phi^k |r_0|$ 을 만족하게 되면 위의 CG 반복수행을 멈춘다.

식(17), (18)에서  $\widehat{H}^k z_m$ 와  $\widehat{H}^k u_m$ 의 계산은 다음을 이용한다.

$$v = \widehat{H}_k \Delta f \quad (20)$$

$P_w$ 에 대응되는 벡터  $\Delta f$ 에 대해 어떤 벡터  $v$ 가 식(20)과 같이 주어질 때, 벡터  $v$ 를 구하기 위해 Bertsekas and Gafni<sup>3)</sup>에서는 다음과 같은 간단식을 제안하였다. 여기서  $v(p)$ 는 벡터  $v$ 의 경로  $p$ 성분이며  $\Delta y_a$ 는 벡터  $\Delta f$ 에 대응되는 링크 성분이다.

$$\begin{aligned} v(p) &= \sum_{\text{link } a \in p} S_a' \Delta y_a - \sum_{\text{link } a \in \overline{p}_w} S_a' \Delta y_a \\ \forall p \in P_w, p \in C^k, p \neq \overline{p}_w, w \in W \end{aligned} \quad (21)$$

이를 이용하면 식(15)~(19)의 계산들은 결국 다음과 같은 방법으로 구할 수 있다. 여기서  $x(p)$ 는 벡터  $x$ 에서 경로  $p$  성분을 의미한다.

$$z_0(p) = 0 \quad (22)$$

$$z_{m+1}(p) = z_m(p) + \gamma_m u_m(p), \quad m=0,1,\dots \quad (23)$$

$$u_0(p) = - \sum_{a \in L_p} S_a' \cdot r_0(p) \quad (24)$$

$$\begin{aligned} u_m(p) &= - \sum_{a \in L_p} S_a' r_m(p) + \beta_m u_{m-1}(p) \\ & \quad m=1,2,\dots \end{aligned} \quad (25)$$

$$\begin{aligned} r_m(p) &= \sum_{a \in p} S_a' \cdot z_m(p) + (t_p - t_{p_w}) \\ & \quad m=0,1,\dots, \end{aligned} \quad (26)$$

$$\begin{aligned} \gamma_m &= \frac{\sum_p \left\{ \sum_{a \in L_p} S_a' \cdot r_m^2(p) \right\}}{\sum_p \left\{ \sum_{a \in p} S_a' \cdot u_m^2(p) \right\}} \quad m=0,1,\dots \\ & \quad m=0,1,\dots \end{aligned} \quad (27)$$

$$\begin{aligned} \beta_m &= \frac{\sum_p \left\{ \sum_{a \in L_p} S_a' \cdot r_m^2(p) \right\}}{\sum_p \left\{ \sum_{a \in L_p} S_a' \cdot r_{m-1}^2(p) \right\}} \\ & \quad m=0,1,\dots \end{aligned} \quad (28)$$

### 3. 통행배정 알고리즘의 단계

앞에서 설명된 내용을 종합하여 알고리즘을 정리하면 다음과 같다.

#### • MAIN STEP

1. All-or-Nothing 배정
2. 각 O/D 쌍에 대해 최단 경로를 구한다.
3.  $C^k$ 를 구한다.
4. 모든 경로에 대해 다음을 실시한다.

경로  $p$ 가  $C^k$ 에 속하지 않으면,

$$f_p^{k+1} = [f_p^k - \alpha \overline{H}_p^{-1} (t_p^k - t_{p_w}^k)]^+$$

$C^k$ 에 속하면, CG 반복수행(subroutine CG)을 이용,  $\widehat{d}^k$ 를 구한다.

$$f_p^{k+1} = [f_p^k + \bar{a} \hat{d}^k]^+$$

5. 최단 경로에 다음을 배정한다.

$$f_{p_v}^{k+1} = r_w - \sum_{\substack{p \in L_p \\ p \neq \bar{p}_v}} f_p^{k+1}$$

6. 수렴하지 않으면 2로 가고, 수렴하면 끝낸다.

• Subroutine CG

1.  $z_0=0$ 을 정의하고, 이를 이용하여  $r_0$ 과  $u_0, \gamma_0$ 을 구한다.

$$r_0(p) = t_p - t_{\bar{p}_v}$$

$$u_0(p) = - \sum_{a \in L_p} S'_a \cdot r_0(p)$$

$$\gamma_0 = \frac{\sum_p \left\{ \sum_{a \in L_p} S'_a \cdot r_0^2(p) \right\}}{\sum_p \left\{ \sum_{a \in L_p} S'_a \cdot u_0^2(p) \right\}}$$

2. 다음 식을 이용하여,  $z_1$ 을 구한다.

$$z_1 = z_0 + \gamma_0 u_0$$

3. 다음 식을 이용하여  $r_m, \beta_m, u_m, \gamma_m$ 를 구한다.

$$r_m(p) = \sum_{a \in p} S'_a \cdot z_m(p) + (t_p - t_{\bar{p}_v}) \quad m=1,2, \dots,$$

$$\beta_m = \frac{\sum_p \left\{ \sum_{a \in L_p} S'_a \cdot r_m^2(p) \right\}}{\sum_p \left\{ \sum_{a \in L_p} S'_a \cdot r_{m-1}^2(p) \right\}} \quad m=1,2, \dots$$

$$u_m(p) = - \sum_{a \in L_p} S'_a r_m(p) + \beta_m u_{m-1}(p)$$

$$m=1,2, \dots$$

$$\gamma_m = \frac{\sum_p \left\{ \sum_{a \in L_p} S'_a \cdot r_m^2(p) \right\}}{\sum_p \left\{ \sum_{a \in L_p} S'_a \cdot u_m^2(p) \right\}} \quad m=1,2, \dots$$

4.  $z_{m+1}$  갱신.

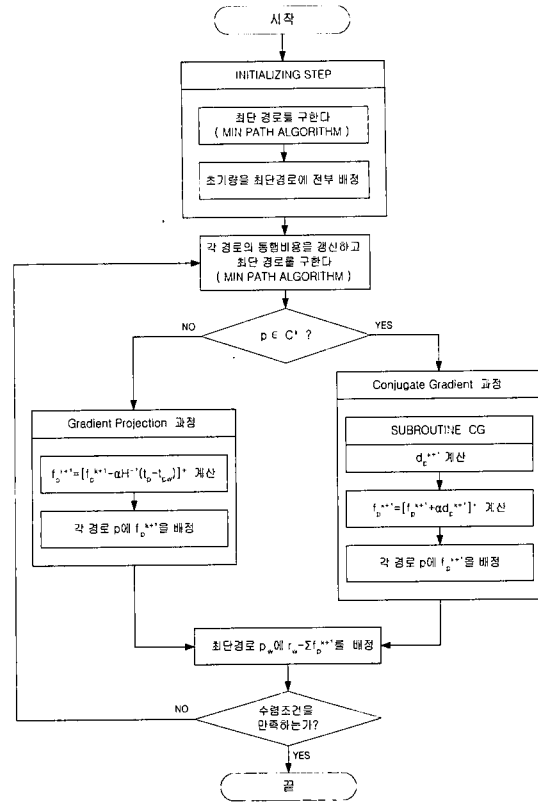
$$z_{m+1} = z_m + \gamma_m u_m, \quad m=1,2, \dots,$$

5. 수렴조건을 만족하면 Subroutine CG를 끝내고 그렇지 않으면, Step 3으로 간다.

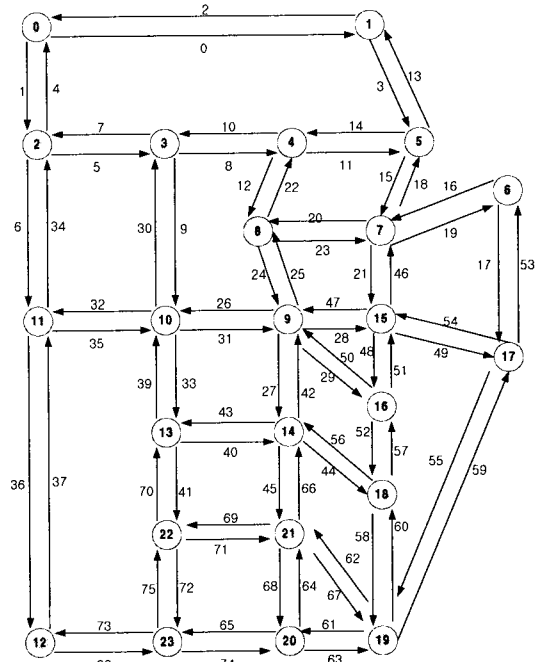
N. 알고리즘의 적용

CG 를 이용한 경로기반 통행배정 알고리즘의 적용은 LeBlanc<sup>7)</sup>이 제시한 Sioux Falls Network와 O/D 자료를 이용하였다. 이 Network는 24개의 노드와 76개의 링크로 구성되어 있으며, 그 형태는 <그림 2>에 나타나 있다.

O/D 자료는 <표 1>에서 나타난 바와 같으며, 총 528개의 대칭적인 O/D쌍들로 이루어져 있고 링크의



<그림 1> 경로기반통행배정 알고리즘 순서도



<그림 2> Sioux Falls Network

<표 1> 알고리즘의 적용에 쓰인 O/D 자료 ( ×10<sup>3</sup> unit)

D \ O	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	1	1	5	2	3	5	8	5	13	5	2	5	3	5	5	4	1	3	3	1	4	3	1
1	1	0	1	2	1	4	2	4	2	6	2	1	3	1	1	4	2	0	1	1	0	1	0	0
2	1	1	0	2	1	3	1	2	1	3	3	2	1	1	1	2	1	0	0	0	0	1	1	0
3	5	2	2	0	5	4	4	7	7	12	14	6	6	5	5	8	5	1	2	3	2	4	5	2
4	2	1	1	5	0	2	2	5	8	10	5	2	2	1	2	5	2	0	1	1	1	2	1	0
5	3	4	3	4	2	0	4	8	4	8	4	2	2	1	2	9	5	1	2	3	1	2	1	1
6	5	2	1	4	2	4	0	10	6	19	5	7	4	2	5	14	10	2	4	5	2	5	2	1
7	8	4	2	7	5	8	10	0	8	16	8	6	6	4	6	22	14	3	7	9	4	5	3	2
8	5	2	1	7	8	4	6	8	0	28	14	6	6	6	9	14	9	2	4	6	3	7	5	2
9	13	6	3	12	10	8	19	16	28	0	40	20	19	21	40	44	39	7	18	25	12	26	18	8
10	5	2	3	15	5	4	5	8	14	39	0	14	10	16	14	14	10	1	4	6	4	11	13	6
11	2	1	2	6	2	2	7	6	6	20	14	0	13	7	7	7	6	2	3	4	3	7	7	5
12	5	3	1	6	2	2	4	6	6	19	10	13	0	6	7	6	5	1	3	6	6	13	8	7
13	3	1	1	5	1	1	2	4	6	21	16	7	6	0	13	7	7	1	3	5	4	12	11	4
14	5	1	1	5	2	2	5	6	10	40	14	7	7	13	0	12	15	2	8	11	8	26	10	4
15	5	4	2	8	5	9	14	22	14	44	14	7	6	7	12	0	28	5	13	16	6	12	5	3
16	4	2	1	5	2	5	10	14	9	39	10	6	5	7	15	28	0	6	17	17	6	17	6	3
17	1	0	0	1	0	1	2	3	2	7	2	2	1	1	2	5	6	0	3	4	1	3	1	0
18	3	1	0	2	1	2	4	7	4	18	4	3	3	3	8	13	17	3	0	12	4	12	3	1
19	3	1	0	3	1	3	5	9	6	25	6	5	6	5	11	16	17	4	12	0	12	24	7	4
20	1	0	0	2	1	1	2	4	3	12	4	3	6	4	8	6	6	1	4	12	0	18	7	5
21	4	1	1	4	2	2	5	5	7	26	11	7	13	12	26	12	17	3	12	24	18	0	21	11
22	3	0	1	5	1	1	2	3	5	18	13	7	8	11	10	5	6	1	3	7	7	21	0	7
23	1	0	0	2	0	1	1	2	2	8	6	5	7	4	4	3	3	0	1	4	5	11	7	0

<표 2> Sioux Falls Network 링크 특성

#	from	to	T <sub>0</sub>	C	α	β	#	from	to	T <sub>0</sub>	C	α	β	#	from	to	T <sub>0</sub>	C	α	β	#	from	to	T <sub>0</sub>	C	α	β
0	0	1	0.05	2000	0.77	1.2	19	7	6	0.03	2000	0.77	1.2	38	12	23	0.03	2000	0.77	1.2	57	18	16	0.03	2000	0.93	1.8
1	0	2	0.03	2000	0.77	1.2	20	7	8	0.13	2000	0.93	1.8	39	13	10	0.05	2000	0.93	1.8	58	18	19	0.05	2000	0.93	1.8
2	1	0	0.05	2000	0.77	1.2	21	7	15	0.06	2000	0.93	1.8	40	13	14	0.06	2000	0.93	1.8	59	19	17	0.03	2000	0.77	1.2
3	1	5	0.04	2000	0.77	1.2	22	8	4	0.06	2000	0.93	1.8	41	13	22	0.05	2000	0.93	1.8	60	19	18	0.05	2000	0.93	1.8
4	2	0	0.03	2000	0.77	1.2	23	8	7	0.13	2000	0.93	1.8	42	14	9	0.08	2000	0.93	1.8	61	19	20	0.05	2000	0.77	1.2
5	2	3	0.05	2000	0.93	1.8	24	8	9	0.04	2000	0.93	1.8	43	14	13	0.06	2000	0.93	1.8	62	19	21	0.06	2000	0.93	1.8
6	2	11	0.03	2000	0.77	1.2	25	9	8	0.04	2000	0.93	1.8	44	14	18	0.04	2000	0.93	1.8	63	20	19	0.05	2000	0.77	1.2
7	3	2	0.03	2000	0.93	1.8	26	9	10	0.06	2000	0.93	1.8	45	14	21	0.04	2000	0.93	1.8	64	20	21	0.03	2000	0.93	1.8
8	3	4	0.03	2000	0.93	1.8	27	9	14	0.08	2000	0.93	1.8	46	15	7	0.06	2000	0.93	1.8	65	20	23	0.03	2000	0.77	1.2
9	3	10	0.08	2000	0.93	1.8	28	9	15	0.05	2000	0.93	1.8	47	15	9	0.05	2000	0.93	1.8	66	21	14	0.04	2000	0.93	1.8
10	4	3	0.03	2000	0.93	1.8	29	9	16	0.10	2000	0.93	1.8	48	15	16	0.03	2000	0.93	1.8	67	21	19	0.06	2000	0.93	1.8
11	4	5	0.05	2000	0.93	1.8	30	10	3	0.08	2000	0.93	1.8	49	15	17	0.04	2000	0.93	1.8	68	21	20	0.03	2000	0.93	1.8
12	4	8	0.06	2000	0.93	1.8	31	10	9	0.06	2000	0.93	1.8	50	16	9	0.10	2000	0.93	1.8	69	21	22	0.05	2000	0.93	1.8
13	5	1	0.04	2000	0.77	1.2	32	10	11	0.08	2000	0.93	1.8	51	16	15	0.03	2000	0.93	1.8	70	22	13	0.05	2000	0.93	1.8
14	5	4	0.05	2000	0.93	1.8	33	10	13	0.05	2000	0.93	1.8	52	16	18	0.03	2000	0.93	1.8	71	22	21	0.05	2000	0.93	1.8
15	5	7	0.02	2000	0.77	1.2	34	11	2	0.03	2000	0.77	1.2	53	17	6	0.02	2000	0.77	1.2	72	22	23	0.03	2000	0.93	1.8
16	6	7	0.03	2000	0.77	1.2	35	11	10	0.08	2000	0.93	1.8	54	17	15	0.04	2000	0.93	1.8	73	23	12	0.03	2000	0.77	1.2
17	6	17	0.02	2000	0.77	1.2	36	11	12	0.03	2000	0.77	1.2	55	17	19	0.03	2000	0.77	1.2	74	23	20	0.03	2000	0.77	1.2
18	7	5	0.02	2000	0.77	1.2	37	12	11	0.03	2000	0.77	1.2	56	18	14	0.04	2000	0.93	1.8	75	23	22	0.03	2000	0.93	1.8

성능 함수는 일반적으로 사용되고 있는 BPR함수를 사용하였으며 각 링크의 특성은 <표 2>와 같다.

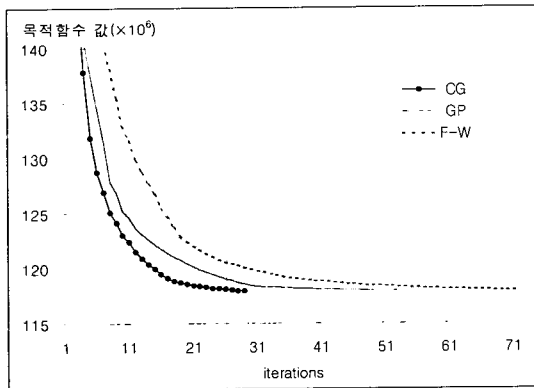
본 연구에서는 이 자료들에 대하여 각각 F-W 알고리즘, GP 알고리즘과 CG 알고리즘의 세 가지 방법을 이용하여 그 결과를 비교하였으며, 비교대상은 목적 함수의 값과 수렴하기까지의 반복수행 횟수, 처리 시간을 선정하였다. 또한, 경로기반 배정 알고리즘

인 GP와 CG의 경우에는 사용된 경로를 모두 나타내기 어려우므로 사용된 경로의 수를 표시하였다.

F-W 알고리즘을 이용한 통행배정은 line search에는 최적화기법 중 Bisection 알고리즘을 사용하였고, GP 알고리즘의 경우에는 이동간격으로 0.05의 상수를 사용하였고, ε=0.3으로하여 경로 교통량의 예상 변화율이 0.3% 이하가 되는 경우에는 CG 과정

〈표 3〉 알고리즘 적용 결과의 비교

Conjugate Gradient			Gradient Projection			Frank-Wolfe		
iteration	목적함수값	처리시간 (sec)	iteration	목적함수값	처리시간 (sec)	iteration	목적함수값	처리시간 (sec)
0	240,867,136	0.07	0	240,867,136	0.07	0	240,825,248	0.07
4	131,857,072	0.26	6	131,490,744	0.30	10	131,575,128	0.55
11	121,508,968	0.63	15	121,750,176	0.68	21	121,710,112	1.10
19	118,566,576	1.05	28	118,567,296	1.24	44	118,570,744	2.25
28	117,925,136	1.52	52	117,927,064	2.26	71	117,934,368	3.60
64	117,647,944	3.40	109	117,676,696	4.71	183	117,705,992	9.24
사용된 경로의 수		1431	사용된 경로의 수		1431			



〈그림 3〉 알고리즘 수렴과정 비교

으로, 0.3% 이상이 되는 경로에 대해서는 GP 과정으로 계산하였다. CG 알고리즘을 이용한 수행에서는 이동간격으로 0.05을 사용하였고, Subroutine CG의 최대 반복수행 횟수는 10회,  $\delta_k=0.01$ 을 사용하여 Subroutine CG에서  $z_m$ 의 반복수행 과정 중 반복수행 횟수가 10회가 초과하거나, 그 변화율이 0.01 미만이 되었을 경우에 반복수행을 끝마치도록 하였다. 한편 전체 알고리즘의 수렴조건은 목적함수의 변화율이 0.001%이내에 들도록 하였다.

각각의 알고리즘을 이용하여 통행 배정을 실시한 결과는 〈표 3〉에 나타나 있다. 이 결과에 따르면, CG 알고리즘은 F-W 알고리즘에 비해 반복수행 횟수에서는 65.6%, 처리 시간의 경우는 63%정도 단축된 것으로 나타났다. GP 알고리즘에 비해서는 반복수행 횟수에서 41.2%, 처리시간에서 27.7%정도 단축된 것으로 나타났다. 여기서, 반복수행 횟수가 단축된 폭보다 처리시간이 단축된 폭이 작은 것은 CG 알고리즘의 경우 보다 정확한 이동간격을 찾기 위해 추가적인 순환 계산이 이루어지기 때문이다. 계산에 사용된

경로의 수는 CG와 GP 알고리즘이 모두 1,431개가 사용되었다. 이는 각 O/D 쌍마다 평균 2.7개 정도가 사용되었음을 알 수 있다.

〈그림 2〉에서는 알고리즘의 수렴과정이 나타나 있다. 이 그림은 〈표 3〉에서 CG 결과의 30번째 반복수행에서의 목적함수 값과 같은 값에 도달하기까지의 각 알고리즘의 진행 과정이다. 여기서 보면 CG에서의 목적 함수의 감소가 F-W와 GP에 비해 현저히 빠르게 진행됨을 알 수 있다. 또한 CG의 감소 패턴이 GP와 다소 비슷함을 알 수 있는데 이는 CG가 일부 경로에서는 GP에서의 이동과 같은 방법으로 최적해로 접근하기 때문이다.

### V. 결론 및 향후 연구과제

본 연구에서는 경로기반 통행배정 알고리즘의 하나인 GP 알고리즘의 최적화에 근처의 수렴속도 문제를 극복하기 위해 기존의 GP 알고리즘에 CG 법을 결합시켜 보다 효율적인 경로기반 통행배정 알고리즘을 구축하였다. 즉, 초기에는 GP 알고리즘에서와 같은 방법으로 진행하다가 최적해 근처에서는 CG 법을 이용, 정확한 이동간격을 찾아 진행하여 빠른 시간 내에 해를 찾을 수 있도록 구성되었다.

또한, 새로운 알고리즘을 가로망에 적용한 결과, 수렴 속도와 목적함수의 최소화에서 기존의 알고리즘들보다 뛰어난 결과를 보여주었다. 따라서, 이 알고리즘은 실시간 동적 통행배정에 이용될 때 많은 이익을 제공할 수 있을 것으로 예상된다.

그러나, 이러한 경로기반 통행배정 알고리즘이 범용적으로 사용되기 위해서는 다음과 같은 사항들에 대한 좀 더 깊이 있는 연구가 추가적으로 이루어져야 할 것이다.



첫째, 최적의 효율성을 나타낼 수 있는  $\epsilon$ 나  $\delta_k$ 값의 선정에 대한 연구가 필요하다. 특히 수렴성을 보장하며, 동시에 충분한 효율성을 가질 수 있는 이동간격  $\bar{\alpha}$ 의 선정에 대한 연구는 앞으로 계속되어야 할 것이다.

둘째, 이러한 새로운 통행배정 알고리즘을 이용한 좀 더 효율적인 실시간 동적 통행배정 알고리즘에 관한 연구도 이루어져야 할 것이다. 빠른 시간 내에 해를 도출해야하는 동적 통행배정의 경우에 정적 통행배정에서의 이러한 효율성의 향상은 큰 이익을 가져다 줄 수 있을 것이다.

**참고문헌**

1. Avriel, M., Nonlinear Programming : Analysis and Methods, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1976.
2. Bertsekas, D. P., "Projected Newton Methods for Optimization Problems with Simple Constraints," SIAM J. Control and Optimization, Vol. 20, 1982.
3. Bertsekas, D. P., and E. M. Gafni, "Projected Newton Methods and Optimization of Multi-commodity Flows." IEEE Transactions on Automatic Control, Vol. AC-28, No. 12, December 1983.
4. Bertsekas, D. P., and R. Gallager, Data Networks, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1992.
5. Chen, A., D-H. Lee, and R. Jayakrishnan, "Path-based Algorithms for Large Scale Traffic Equilibrium Problem : A Comparison between DSD and GP," submitted to Transportation Science, 1999.
6. Jayakrishnan, R., Wei K. Tsai, Joseph N. Prashker, and Subodh Rajadhyaksha, "Faster Path-Based Algorithm for Traffic Assignment," Transportation Research Record 1443, 1994.

7. LeBlanc, L. J., E. K. Morlok, and W. Pierskalla, "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem," Transportation Research 9(5), 1975.
8. Luenberger, D. G., Introduction to Linear and Nonlinear Programming, Addison-Wesley Publishing Company, Reading, MA., 1973.
9. Luenberger, D. G., Linear and Nonlinear Programming, Addison-Wesley Publishing Company, Reading, MA., 1984.
10. Oppenheim, N., Urban Travel Demand Modeling, John Wiley & Sons, Inc., New York, 1995.
11. Ortega, J. M. and W. C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, Inc., San Diego, CA., 1970.
12. Patrikson, M., The Traffic Assignment Problem : Models and Methods, VSP Utrecht, 1994.
13. Sheffi, Y., Urban Transportation Networks, Prentice-Hall, Inc., Englewood Cliffs, N. J., 1985.
14. Sun, Carlos, R. Jayakrishnan, and Wei K. Tsai, "Computational study of a path-based algorithm and its variants for static traffic assignment," Transportation Research Record 1537, 1996.
15. 권용석, "통행료체계에서의 경로기반 통행배정 모형 개발과 적용에 관한 연구", 박사학위논문, 서울대학교, 1999.

✉ 주 작 성 자 : 강승모  
 ✉ 논문투고일 : 2000. 8. 2  
 논문심사일 : 2000. 9. 4 (1차)  
 : 2000. 9. 28 (2차)  
 심사완료일 : 2000. 9. 28