■ 論 文 ■

Bi-level program에서 Cournot-Nash게임과 Stackelberg게임의 비교연구

Comparison between Cournot-Nash and Stackelberg Game in Bi-level Program

임 용 택
(여수대학교 교통물류시스템공학부 조교수)

임 강 원
(서울대학교 환경대학원 교수)

목 차

Key Words : Cournot-Nash게임, Stackelberg게임, Bi-level program, 민감도, 연속형 교통망설계문제

요 약

본 연구에서는 바이레벨 문제를 풀기 위한 2가지 접근법, 즉 Cournot-Nash게임과 Stackelbgerg게임을 서로 비교하기 위한 것으로, 하위문제가 결정적인 통행배정문제(deterministic traffic assignment)인 경우와 확률적 통행배정문제(stochastic traffic assignment)인 경우로 구분하여 분석한다. 바이레벨 프로그램(bi-level program)은 상위문제(upper level program)과 하위 문제(lower level program)로 구성된 수리적인 문제로 상위문제는 목적하는 특정함수를 최적화시키는 형태이며, 하위문제는 통행자의 행태를 반영하는 형태로 구축된다.

기존에 제시된 알고리듬중 바이레벨문제의 대표적인 풀이 알고리듬인 IOA(Iterative Optimization Assignment) 알고리듬과 기종점 통행행렬추정(OD matrix estimation)에 주로 사용되는 IEA(Iterative Estimation Assignment)은 상위문제와 하위문제가 서로 독립적으로 존재하면서 설계변수와 통행량을 서로 주고 받는 형태를 갖고 있어 Cournot-Nash게임형태이다. 이에 반해, 최근에 제시된 민감도분석(Sensitivity analysis)을 기초로 한 알고리듬들은 상위문제에서 결정된 설계변수 변화에 대해 하위문제의 통행량변화를 민감도를 통해 고려하기 때문에 Stackelbeg게임이라고 볼 수 있다. 본 연구에서는 이들 알고리듬들을 비교하는 데 연구의 목적이 있으며, 기존에 제시된 기법과는 다른 좀 더 효율적인 접근법을 제시한다. 예제 교통망을 이용하여 제시된 모형들을 비교해본 결과, 결정적인 통행배정모형을 하위문제로 설정한 경우에는 두가지 접근법 모두 동일한 상위목적함수 값을 보여 우위를 판단할 수 없었지만, 확정적 통행배정모형으로 설정한 경우, Stackelberg게임 접근법이 Cournot-Nash게임 접근법 보다 더 우수함을 확인할 수 있었다.

I. 서론

바이레벨 프로그램(bi-level program)은 상위문제(upper level program)과 하위 문제(lower level program)로 구성된 수리적인 문제로 최근 교통분야에 활발히 적용되고 있다. 바이레벨 문제는 도로 확장, 최적 교통신호설계, 혼잡통행료부과 그리고 교통정보제공 등 교통혼잡을 완화시키기 위한 제반 교통정책들을 평가하기 위한 연구분야에 이용되어 왔는데, 상위문제는 목적하는 특정함수를 최적화시키는 형태이며, 하위문제는 통행자의 행태를 반영하는 형태로 구축된다. 그런데, 상위문제와 하위문제를 구성할 때 각 문제들이 서로 협력없이(Noncooperative) 자신들만의 목적을 최적화시키는 형태로 구성하느냐, 아니면 리더(leader)와 추종자(follower)가 존재하여, 리더는 추종자의 행태를 알 수 있다는 가정하에 문제를 구성하느냐에 따라 모형식이 달라지게 된다. 전자를 Cournot-Nash게임이라고 하고 후자를 Stackelberg게임이라고 한다. 이들 게임에 대해서는 Fisk(1984), Yang et al.(1998)등에 잘 설명되어 있다.

앞에서 기술한 도로확장이나, 교차로 신호시간결정, 혼잡통행료 부과 등 여러 교통정책의 경우, 일반적으로 교통운영자(리더)들은 이들 정책이 도입되었을 때 운전자(추종자)들이 어떻게 그들의 경로를 변경할 것인지를 알고 있다고 가정하는 것이 타당하기 때문에 Stackelberg게임이 된다. 즉, 교통분야에서 다루고자하는 제반 교통정책들은 교통운영자가 통행자의 행태를 알고 교통정책변수를 결정한다고 가정하는 것이 현실적이기 때문에, Cournot-Nash게임 보다는 Stackelberg게임에 가깝다고 할 수 있다. 그러나, 현재까지 제시된 대부분의 bi-level 문제들이 Cournot-Nash게임형태로 구성되어 풀고 있는데, 이는 Stackelberg게임으로 구성할 경우 풀기가 어렵기 때문이다.

본 연구에서는 바이레벨 문제를 풀기 위한 2가지 접근법, 즉 Cournot-Nash게임과 Stackelbgerg게임을 서로 비교하기 위한 것으로, 하위문제가 결정적인 통행배정문제(deterministic traffic assignment)인 경우와 확률적 통행배정문제(stochastic traffic assignment)인 경우로 구분하여 분석한다. 바이레벨문제에서 결정적 통행배정문제를 Stackelberg게임으로 해석한 연구는 Tobin et al.(1988), Yang(1995), Yang(1997) 등이 있으나, 풀이과정이 복잡하다는 한계가 있다. 본 연구에서는 이들과는 다른 좀 더 효율적인 접근법을 제시한다. 확률적인 통행배정문제를 Stackelberg게임으로 해석한 연구는 최근에야 제시되었으며(임용택, 2004), 본 연구에서도 이 방법을 채택하여 두 게임이론간의 차이를 분석한다. 다음 절에서는 Cournot-Nash게임과 Stackelberg게임의 차이에 대하여 기술하고, 제Ⅲ절에서는 결정적 형태와 확률적 형태로 구성된 모형식과 알고리듬을 기술하며, 제Ⅳ절에서는 개발된 모형들을 단순 예제교통망을 대상으로 비교평가한다.

Ⅱ. Cournot-Nash게임과 Stackelberg게임

Bi-level 프로그램에서 상위문제와 하위문제가 서로 협력없이(Noncooperative) 자신들만의 목적을 최적화시키는 형태로 구성하면 Counot-Nash게임이 되며, 리더(leader)와 추종자(follower)가 존재하고, 리더는 추종자의 행태를 알 수 있다는 가정하에 문제를 구성하면 Stackelberg게임이 된다. Fisk(1984)는 이들 게임의 차이를 다음과 같이 설명하였다.

만약 2명의 플레이어(player)가 존재하며, 각 플레이어의 목표는 자신의 목적함수 P_1, P_2를 최소화시키는 것으로 가정하면, Cournot-Nash게임의 비협력상황은 균형 상태에서 어떤 플레이어도 자신의 결정을 일방적으로 바꿈으로써 자신의 목적을 향상시킬 수 없다는 특징을 가지고 있다. 따라서 각 플레이어의 결정 변수인 x_1, x_2가 균형상태의 해(x_1^*, x_2^*) 라면 다음과 같은 조건을 만족하게 된다.

$$P_1(x_1, x_2^*) \geq P_1(x_1^*, x_2^*)$$
$$P_2(x_1^*, x_2) \geq P_2(x_1^*, x_2^*) \qquad (1)$$

다른 플레이어의 전략이 주어진 상태에서, i 플레이어의 최적 전략은 다음의 식을 푸는 해법을 찾음으로써 발견될 수 있다.

$$\min_{x_i} \; P_i(x_1, x_2) \qquad (2)$$

그리고 균형 해는 식(2)의 최적해 조건을 둘 다(for $i=1,2$) 동시에 만족시키는 해이다.

이에 반해, Stackelberg 게임에서는 한명의 플레이어(리더)는 다른 플레이어(추종자)가 자신의 결정에 대

해서 어떻게 반응할지 알고 있다. 만약 플레이어 1이 리더이고 $x_2 = T(x_1)$ 가 리더의 결정 x_1 에 대한 플레이어2의 반응이라면, 균형 상태는 다음과 같다.

$$\begin{aligned} P_1(x_1, T(x_1)) &\geq P_1(x_1^*, T(x_1^*)) \\ P_2(x_1, T(x_1)) &\leq P_2(x_1, x_2) \end{aligned} \tag{3}$$

플레이어1의 전략 x_1 에 대하여 플레이어 2의 최적 전략은 다음과 같이 구해진다.

$$\min_{x_2} \quad P_2(x_1, x_2) \tag{4}$$

반면, 리더(leader)의 최적 전략은 다음 식(5)를 푸는 것으로 정리된다.

$$\min_{x_1} \quad P_1(x_1, T(x_1)) \tag{5}$$

즉, Cournot-Nash게임의 해를 구하는 식(2)와 Stackelberg게임 해를 구하는 식(4) 및 식(5)사이에는 분명한 차이가 있음을 알 수 있다. Fisk는 최소화문제를 통하여 Stackelberg게임이 Cournot-Nash게임보다 더 좋은 해를 도출한다는 사실을 보여주었다. 이것은 상위문제인 리더가 하위문제인 추종자의 움직임을 알고 있다는 상황(cooperative)하에서 문제를 풀기 때문인데, 이때 상위문제와 하위문제간에는 명시적인 함수형태가 존재하게 된다.

기존에 제시된 알고리듬들을 게임측면에서 살펴보면, 교통망설계문제의 대표적인 풀이 알고리듬인 IOA (Iterative Optimization Assignment) 알고리듬과 기종점 통행행렬추정(OD matrix estimation)에 주로 사용되는 IEA(Iterative Estimation Assignment)은 상위문제와 하위문제가 독립적으로 존재하면서 설계변수와 통행량을 서로 주고 받는 형태를 갖고 있어 Cournot-Nash게임형태임에 반해, 최근에 제시된 SAB알고리듬 (Sensitivity Analysis Based Algorithm; Yang, 1995)은 상위문제에서 결정된 설계변수 변화에 대해 하위문제의 통행량변화를 민감도를 통해 고려하기 때문에 Stackelbeg게임이라고 볼 수 있다.

Ⅲ. Bi-level program과 풀이과정

1. Bi-level program의 구조

본 연구에서 비교코자 하는 모형은 모두 4가지로 결정적 모형과 확률적모형에 대하여 각각 Cournot-Nash 게임과 Stackelberg게임으로 나누어 모형을 평가한다. 먼저, 본 연구에서 사용되는 주요 변수는 다음과 같다.

- x : 링크 교통량 벡타 $\{\ldots, x_a, \ldots\}$, $a \in A$
- c : 링크 통행비용 벡타 $\{\ldots, c_a, \ldots\}$, $a \in A$
- $p(c)$: 링크 선택확률벡타 $\{\ldots, p_a^{rs}(c), \ldots\}$, $a \in A$, $rs \in F$
- T : OD 통행수요 벡타 $\{\ldots, T_{rs}, \ldots\}$, $rs \in F$
- s : 설계변수 벡타 $\{\ldots, s_b, \ldots\}$, $b \in B$
- $u(s)$: 설계변수에 대한 건설비용벡타 $\{\ldots, u_b(s), \ldots\}$, $b \in B$

여기서, A는 모든 링크집합이며, B는 설계변수가 고려되는 링크집합을 나타낸다. 또한, F는 모든 기종점 집합이다.

바이레벨 모형의 상위문제는 여러 형태로 구성될 수 있는데, 교통망설계문제(Network design problem)의 경우 주로 사용되는 목적함수는 건설비용을 포함한 총통행비용을 최소화시키는 문제로 구성되어 왔다. 여기서 건설비용이란 모형에서 도출된 최적 설계변수만큼 교통망을 변화시키는데 소요되는 비용(예를 들어 용량을 확장시키기 위한 설치비용 등)을 의미한다. 본 연구에서도 총통행비용 최소화를 상위문제로 구성하며, Cournot-Nash게임과 Stackelberg게임은 다음과 같이 표현된다.

[상위문제1] Cournot-Nash게임 형태

$$min_s \; Z(s, x) = \sum_{a \in A} x_a \cdot c_a(x_a, s) + \sum_{b \in B} u_b(s) \tag{6}$$

[상위문제2] Stackelberg게임 형태

$$min_s \; Z(s, x(s)) = \sum_{a \in A} x_a(s) \cdot c_a(x_a(s), s) + \sum_{b \in B} u_b(s) \tag{7}$$

여기서, x와 c는 링크 통행량벡타와 링크 통행비용벡타이고 s는 설계변수(design parameter)이며, $u(s)$는 건설비용벡타이다. 상위문제에서 유의할 점은 Stackelberg게임의 경우, 링크 통행벡타 x가 설계변수 s의 함수라는 것이다. 즉, 이것은 설계변수가 변함에 따라 도출되는 링크통행량의 변화를 상위문제에 고려한다는 것으로, 이는 리더가 추종자의 변화를 알고 상위문제

를 결정한다는 Stackelberg게임형태로 상위문제가 구성되었음을 의미한다. 즉, 교통망설계문제에서 일반적으로 사용되는 상위문제인 식(6)과 같은 Cournot-Nash 게임과 식(7)의 Stackelberg게임사이에는 모형식에 차이가 있다.

하위문제의 경우, 변경된 교통망에 따라 변화된 통행자의 경로선택행위를 반영해야 하기 때문에 통행배정(traffic assignment)모형이 하위문제가 된다. 본 연구에서는 결정적 통행배정문제(deterministic traffic assignment)와 확률적 통행배정문제(stochastic traffic assignment)로 나누어 모형을 구축하며 각 모형은 다음과 같다.

[하위문제1] 결정적 통행배정모형

$$\min\ z(x) = \sum_a \int_0^{x_a} c_a(w)\, dw \tag{8}$$

subject to,

$$\sum_k f_k^{rs} = T_{rs}$$

$$f_k^{rs} \geq 0$$

$$x_a = \sum_{rs}\sum_k f_k^{rs}\delta_{a,k}^{rs}$$

[하위문제2] 확률적 통행배정모형

$$x_a - \sum_{rs \in F} p_a^{rs}(c(x,s))\, T_{rs} = 0 \tag{9}$$

subject to,

〔하위문제1〕의 제약조건식과 동일

여기서, f_k^{rs}는 기종점 rs간 경로 k의 통행량이며, $\delta_{a,k}^{rs}$는 rs간 경로 k가 링크 a에 속하면 1, 그렇지 않으면 0인 가변수 이다.

2. Stackelberg게임의 풀이 알고리듬

Stackelberg게임으로 bi-level프로그램을 푸는 경우 문제가 되는 것은 링크 통행벡타 x와 설계변수 s의 관계를 어떻게 고려하느냐에 있는데, 본 연구에서는 민감도분석(sensitivity analysis)을 통하여 이를 반영한다. Stackelberg게임에서 통행벡타 x와 설계변수 s는 식(10)과 같은 관계를 갖고 있다.

$$x = x(s) \tag{10}$$

식(10)을 s^*에서 선형으로 확장하면 다음과 같은 근사식을 얻는다.

$$x = x(s^*) + \frac{\partial x(s^*)}{\partial s}(s - s^*) \tag{11}$$

여기서, $\frac{\partial x(s^*)}{\partial s}$는 민감도(sensitivity)를 나타내며, Stackelberg게임에서는 식(11)을 통하여 하위문제의 반응을 상위문제에 고려하게 된다. 민감도를 이용한 bi-level문제의 해석 알고리듬을 정리하면 다음과 같다. 알고리듬의 기본 원리는 하위문제와 상위문제의 풀이과정을 반복하게 되는데, 설계변수의 변화에 따른 링크 통행량의 변화를 민감도를 통하여 풀이과정에 반영하게 된다.

[단계0] 초기화

반복수 $n=0$, 초기설계변수 s^0

[단계1] $n=n+1$

[단계2] s^{n-1}을 이용하여 하위문제를 풀어 $x(s^{n-1})$ 산출

[단계3] 민감도값 계산후, 다음 식으로 $x(s^n, s^{n-1})$ 계산 :

$$x(s^n, s^{n-1}) = x(s^{n-1}) + \frac{\partial x(s^{n-1})}{\partial s^n}(s^n - s^{n-1})$$

[단계4] 상위문제 $Z[s^n, x(s^n, s^{n-1})]$를 풀어 설계 변수 s^n 결정

[단계5] 수렴성 검토

만약 $|s^n - s^{n-1}| < \varepsilon$ 이면 정지,

그렇지 않으면 (단계1)로 진행

〔단계2〕의 하위문제와 〔단계4〕의 상위문제를 풀기 위한 방법은 여러 가지가 있다. 상위문제의 경우, 다변수 비제약 비선형 최소화문제(Unconstrained nonlinear minimization program in multidimensions)로 Newton-Rapson방법, DFP(Davidon-Fletcher-Powell) 방법, BFGS (Broyden-Fletcher-Goldfarb-Shanno) 방법 등이 있는데, 본 연구에서는 역행렬(inverse matrix)계산과정이 필요 없는 DFP방법을 이용한다. (임용택(2004) 또는 Bazaraa 외(1993) 참조). 하위문제의 경우, 본 연구에서는 2가지 통행배정모형, 즉 결정적 통행배정모형(deterministic traffic assignment)

과 확률적 통행배정모형(stochastic traffic assignment)으로 구분하여 각각 분석한다. 결정적 통행배정모형은 현재 널리 사용되고 있는 Frank-Wolfe방법을 이용하며, 확률적 통행배정모형은 직접 로짓배정법(direct logit loading method)을 이용한다.(자세한 풀이과정은 임용택(2003) 참조).

여기서 문제가 되는 것은 민감도를 어떻게 구하느냐인데, 민감도는 설계변수의 변화에 대해 하위문제로부터 도출되기 때문에 결정적 통행배정모형인 경우와 확률적 통행배정모형인 경우로 나누어 제시한다.

1) 결정적 통행배정인 경우

하위문제로 결정적 통행배정모형을 고려하는 경우, 링크 통행량 x와 설계변수 s간에 명시적인(explicit) 관계식이 없기 때문에 민감도를 구하기가 쉽지 않다. 이와 관련해서, Tobin et al.(1988), Yang (1995) 등이 SAB(sensitivity analysis-based) 알고리듬을 제시하였는데, 계산과정이 복잡하다는 한계가 있다. 본 연구에서는 좀 더 효율적인 방법으로 미분정의를 직접 사용하여 식(12)와 같이 민감도를 구한다.

$$\frac{\partial x(s^*)}{\partial s} = \lim_{\delta s \to 0} \frac{x(s^* + \delta s) - x(s^*)}{\delta s} \qquad (12)$$

이 방법은 설계변수 s^*와 $s^* + \delta s^*$에서 통행배정을 각각 수행하여 산출된 링크 통행량 $x(s^*)$와 $x(s^* + \delta s^*)$의 차이를 δs로 나누어 민감도를 구하게 된다. 따라서 이 방법은 휴리스틱한 방법이라 할 수 있다.

2) 확률적 통행배정인 경우

확률적 통행배정모형으로 식(13)과 같은 로짓모형(logit model)을 이용하게 되면, 링크 통행량 x와 설계변수 s사이에 명확한 함수관계가 존재한다. 따라서, 정확한 민감도을 구할 수 있다.

$$p_k(c) = \frac{\exp(-\theta c_k)}{\sum_i \exp(-\theta c_i)} \qquad (13)$$

확률적 통행배정문제(식9)로부터 우리는 다음과 같은 새로운 함수를 설정할 수 있다.

$$f(x, s) \equiv x - p(c(x, s))T$$

여기서, $x(s^*)$를 설계변수 s^*가 주어진 상태에서의 확률적 균형통행량이라고 정의하면

$$f(x(s^*), s^*) = 0$$

이 된다. Implicit function theorem에 의해서 민감도는 다음과 같이 구해진다.

$$\frac{\partial x(s^*)}{\partial s} = -\left[\frac{\partial f}{\partial x}\Big|_{(x(s^*), s^*)}\right]^{-1}\left[\frac{\partial f}{\partial s}\Big|_{(x(s^*), s^*)}\right]$$

여기서, $\frac{\partial f}{\partial x}$와 $\frac{\partial f}{\partial s}$는 식(14),(15)와 같으며, 이 식들의 도출과정은 임용택(2004)에 자세히 기술되어 있다.

$$\frac{\partial f_b}{\partial x_a} = \chi_{ba} - \theta\left(\frac{\partial c_a}{\partial x_a}\right)\sum_k [p_k \delta_{ak} - p_k(\sum_i p_i \delta_{ai})]\delta_{bk} \cdot T \qquad (14)$$

$$\frac{\partial f_b}{\partial s_a} = \theta\left(\frac{\partial c_a}{\partial s_a}\right)\sum_k [p_k \delta_{ak} - p_k(\sum_i p_i \delta_{ai})]\delta_{bk} \cdot T \qquad (15)$$

여기서, χ_{ba}는 Kronecker delta이고 $\delta_{ak}, \delta_{bk}, \delta_{ai}$는 $a, b \in k(pathset), a \in i(pathset)$이면 1, 그렇지 않으며 0인 가변수이다.

3. Cournot-Nash게임의 풀이 알고리듬

Cournot-Nash게임의 경우, 상위문제와 하위문제가 서로 상대방에 대한 협력없이 자신들만의 목적함수를 최적화시키는 형태이기 때문에 Stackelberg게임과는 달리 민감도를 통한 설계변수의 영향을 고려할 필요가 없다. 따라서 Cournot-Nash게임에 의한 bi-level문제의 풀이 알고리듬은 다음과 같다.

[단계0] 초기화

반복수 $n=0$, 초기설계변수 s^0

[단계1] $n=n+1$

[단계2] s^{n-1}을 이용하여 하위문제를 풀어 $x(s^{n-1})$ 산출

[단계3] 상위문제 $Z[s^n, x(s^n, s^{n-1})]$를 풀어 설계변수 s^n 결정

[단계4] 수렴성 검토

만약 $|s^n - s^{n-1}| < \varepsilon$ 이면 정지, 그렇지 않으면 (단

계1)로 진행

앞 절에서 기술한 Stackelberg게임에 의한 풀이 알고리듬과의 차이는, Stackelberg게임의 알고리듬중 〔단계3〕, 즉 민감도를 고려하는 과정이 Cournot-Nash게임 알고리듬에는 없다는 점뿐이다.

Ⅳ. 모형의 비교 평가

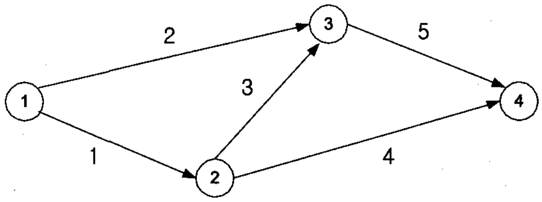
본 연구에서 제시한 풀이 알고리듬은 결정적인 경우와 확률적인 경우로 구분되며, 각 경우에 대하여 Cournot-Nash게임과 Stackelberg게임으로 접근법을 나누어 각 알고리듬들을 비교평가한다.

1. 예제 교통망

알고리듬들을 평가하기 위하여 사용되는 예제 교통망은 〈그림 1〉과 같은 5개의 링크와 3개의 경로로 이루어진 단순한 교통망이다. 기종점은 노드1과 노드4이며 통행수요는 10으로 가정한다. 설계변수는 링크용량이며, 다음과 같은 BPR식을 링크 통행비용함수로 사용한다(링크 속성은 표1 참조).

$$c_a = c_{a0}\left[1+0.15\left(\frac{x_a}{Q_a+s_a}\right)^4\right]$$

따라서, 설계변수를 고려하지 않는 초기상태에서 $s_a=0$이며 모든 링크를 대상으로 용량증가를 고려한다. 설계변수를 수행하는데 소요되는 건설비용함수는 $u_a(s)=20(s_a-0)^2$ 이다. 또한, 로짓모형의 스케일 파라메타는 $\theta=0.1$로 가정한다.



〈그림 1〉 예제교통망

〈표 1〉 링크 속성자료

링크번호	초기통행시간(c_{a0})	용량(Q_a)
1	3	5
2	3	5
3	2	3
4	3	5
5	3	5

2. 분석결과

본 연구에서 제시한 모형들을 예제 교통망에 적용한 결과가 〈표 2〉, 〈표 3〉과 〈표 4〉에 나타나 있다. 〈표 2〉는 4개의 비교모형별로 초기상태와 최종상태의 상위문제 목적함수값 및 건설비용을 보여주고 있다. 먼저 결정적 통행배정모형을 하위문제로 둔 경우, Cournot-Nash게임과 Stackelberg게임 모두 설계변수의 영향이 없는 초기상태 69.0에서 68.844로 동일하게 감소하여 이들 모형간에는 차이가 없었다. 그러나, 확정적 통행배정모형인 경우, Cournot-Nash게임은 81.417에서 80.804로 감소한 반면, Stackelberg게임은 81.417에서 78.904로 감소하였다. 이는 Stackelberg게임이 Counot-Nash게임보다 더 낮은 목적함수값으로 수렴하여 더 좋은 해를 도출하고 있음을 알 수 있다. 이때 건설비용에도 차이가 있음을 알 수 있다. 여기서 확정적 통행배정에서 도출된 상위목적함수 값이 결정적 통행배정모형의 값보다 높게 나타나는 이유는 확정적 통행배정의 경우, 인지오차 비용을 통행비용에 포함하기 때문이다.

〈표 3〉과 〈표 4〉는 각 모형별로 산출된 최적 설계변수값과 각 설계변수별 링크 통행량의 민감도를 나타내고 있다. 표에서 보듯이 각 모형별로 최적 설계변수값에는 서로 약간의 차이가 있다. 여기서 주의해서 볼 것은 민감도 값인데, 민감도는 설계변수의 변화에 따른 링크 통행량의 변화를 나타낸다. 예를 들어 결정적 통행배정인 경우, 링크1의 설계변수(즉, 용량)가 1단위 증가하면 링크1을 이용하는 통행량이 0.2352만큼 증가하게 된다. 그러나 링크3의 경우, 용량을 증가해도 통행량에는 변화가 없는데, 이는 상위목적함수 값을 감소시키는데 기여하지 못한다는 것을 의미한다. 따라서, 링크3의 경우 〈표 3〉에서 보듯이 링크3의 최적 설계변수값은 0으로 나타나 있다. 이에 반해, 확정적 통행배정인 경우에는 양의 민감도를 보여 링크용량의 증가가 통행량을 유도하고 있음을 알 수 있다. 여기서 또하나 관심있는 결과는 〈표 4〉에서 보듯이 링크1과 링크2의 민감도가 각 설계변수값에 대하여 서로 다른 부호를 갖는 동일한 값이라는 점이다. 이것은 예제 교통망의 기하구조가 링크1과 링크2가 서로 보완적인 경로로 작용

〈표 2〉 비교모형별 상위목적함수값과 건설비용

하위문제	접근법	초기		최종	
		상위목적 함수 값	건설 비용	상위목적 함수 값	건설 비용
결정적 통행배정모형	Cournot-Nash 게임	69.000	0.000	68.844	0.148
	Stackelberg 게임	69.000	0.000	68.844	0.148
확률적 통행배정모형	Cournot-Nash 게임	81.417	0.000	80.804	0.682
	Stackelberg 게임	81.417	0.000	78.904	0.404

〈표 3〉 비교모형별 최종 설계변수값

하위문제	접근법	링크 번호	설계변수(s_a)
결정적 통행배정모형	Cournot-Nash게임	1	0.043
		2	0.043
		3	0.000
		4	0.043
		5	0.043
	Stackelberg게임	1	0.043
		2	0.043
		3	0.000
		4	0.043
		5	0.043
확률적 통행배정모형	Cournot-Nash게임	1	0.130
		2	0.009
		3	0.016
		4	0.009
		5	0.129
	Stackelberg게임	1	0.083
		2	0.009
		3	0.010
		4	0.012
		5	0.114

〈표 4〉 설계변수에 대한 각 링크의 민감도

(a) 결정적 통행배정의 경우(Stackelberg게임)

링크 번호	$\frac{\partial x_a}{\partial s_1}$	$\frac{\partial x_a}{\partial s_2}$	$\frac{\partial x_a}{\partial s_3}$	$\frac{\partial x_a}{\partial s_4}$	$\frac{\partial x_a}{\partial s_5}$
1	0.2352	-0.2546	0.0000	0.2352	-0.2546
2	-0.2352	0.2546	0.0000	-0.2352	0.2546
3	0.0000	0.0000	0.0000	0.0000	0.0000
4	0.2352	-0.2546	0.0000	0.2352	-0.2546
5	-0.2352	0.2546	0.0000	-0.2352	0.2546

(b) 확률적 통행배정의 경우(Stackelberg게임)

링크 번호	$\frac{\partial x_a}{\partial s_1}$	$\frac{\partial x_a}{\partial s_2}$	$\frac{\partial x_a}{\partial s_3}$	$\frac{\partial x_a}{\partial s_4}$	$\frac{\partial x_a}{\partial s_5}$
1	0.1057	-0.0182	0.0232	0.0106	-0.0822
2	-0.1057	0.0182	-0.0232	-0.0106	0.0822
3	0.0495	-0.0085	0.0456	-0.0091	0.0701
4	0.0561	-0.0096	-0.0224	0.0197	-0.1524
5	-0.0561	0.0096	0.0224	-0.0197	0.1524

하기 때문에 어떤 하나의 링크 통행량이 증가하면 다른 링크의 통행량은 감소하기 때문이다. 링크4와 링크5도 동일한 패턴을 보이고 있다.

V. 결론

본 연구에서는 바이레벨문제를 풀기 위한 2가지 접근법에 대한 풀이 알고리듬을 제시하고 간단한 예제 교통망을 이용하여 제시된 모형들을 서로 비교하였다. 즉, 기존 대부분의 바이레벨문제들이 이용하고 있는 Cournot-Nash게임 풀이와 본 연구에서 제시한 Stackelberg게임 풀이를 비교하였다. 이를 위하여 상위문제와 하위문제간의 명시적인 함수를 가정하였으며, 민감도분석 기법을 도입하였다. 예제 교통망을 대상으로 평가해 본 결과, 결정적인 통행배정모형을 하위문제로 설정한 경우에는 두가지 접근법 모두 동일한 상위목적함수 값을 보여 우위를 판단할 수 없었지만, 확정적 통행배정모형으로 설정한 경우, Stackelberg게임 접근법이 Cournot-Nash게임 접근법 보다 더 우수함을 확인할 수 있었다. 이런 결과는 Fisk(1984)의 결과와 일치하는 것으로 상위문제가 하위문제의 반응을 모형내에 고려하여 풀기 때문이다. 즉, 서로 협력적인 게임상황하에서 문제를 푼 결과이다. 따라서, 서론에서 언급한 여러 교통정책들에 대한 좀 더 나은 해를 얻기 위해서는 기존 Cournot-Nash게임 형태의 풀이 알고리듬 보다는 Stackelberg게임 형태의 풀이 알고리듬을 적용해야 할 것으로 보인다.

본 연구와 관련된 향후 연구분야로는, 설계변수값의 변화에 따른 통행수요의 변화(elastic demand)를 고려하는 연구와 도로의 확장뿐만 아니라 교차로 신호시간결정, 적정 혼잡통행료 결정, 고속도로 램프미터링률 결정 등 교통운영 관리분야에 적용하는 연구가 있다.

참고문헌

1. 임용택(2003) 확률적 로짓 통행배정모형의 해석 알고리듬, 대한교통학회지, 제21권 제2호, 대한교통학회, pp.95~105.
2. 임용택(2004) 민감도분석을 이용한 연속형 교통망 설계모형의 개발, 대한교통학회지, 제22권 제2호, pp.65~76.
3. Bazaraa,M.S., H.D.Sherall, C.M.Shetty (1993)

Nonlinear programming: theory and algorithms, 2nd ed., Wiley.

4. Fisk,C.S.(1984) Game theory and transportation systems modeling, Transportation Research 18B, pp.301~313.
5. Sheffi Y.(1985) Urban transportation networks, Prentice-Hall.
6. Tobin,R.L., Friesz,T.L.(1988) Sensitivity analysis for equilibrium network flows, Transportation Science 22, pp.242~250.
7. Yang,H.(1995) Heuristic algorithms for the bilevel origin-destination matrix estimation problem, Transportation Research 29B, pp.231~242.
8. Yang H. (1997) Sensitivity analysis for the elastic-demand network equilibrium problem with applications, Transportation Research 31B, pp.55~70.
9. Yang H., M.G.H. Bell(1998) Models and algorithms for road network design: a review and some new developments, Transport Review 18, pp.257~278.

♧ 주 작 성 자 : 임용택
♧ 논문투고일 : 2004. 10. 13
논문심사일 : 2004. 10. 29 (1차)
심사판정일 : 2004. 10. 29
♧ 반론접수기한 : 2005. 4. 30