

□ 論 文 □

다목적 정보 제공을 위한 다경로 탐색 기법 개발

A Multi-path Search Algorithm for Multi-purpose Activities

정 언 정

(서울대학교 지구환경시스템공학부 석사과정)

김 창 호

(美 일리노이 대학교 도시 및 지역계획 시스템 석좌교수)

목 차

- I. 서론
- II. 기존 알고리즘의 고찰
 - 1. 링크 기반 탐색법
 - 2. 다경로 탐색 기법
- III. 링크기반 다목적 다경로 탐색기법
 - 1. 표식
 - 2. 비용계산
- 3. 알고리즘
- IV. 알고리즘의 적용
 - 1. 실험 네트워크의 구성
 - 2. 알고리즘의 적용
 - 3. 알고리즘 결과
- V. 결론
- 참고문헌

Key Words : multi-purpose, multi-path, needs, link-based, 최저비용과의 차이, 유연한 알고리즘 flexible search algorithm

요 약

IT 산업이 발달하고, 정보의 양이 넘쳐날수록, 사람들은 획일화되어 제공되는 정보 보다는, 스스로 다양한 경로를 통해 정보를 찾아내며, 이를 가공하여 판단하고 반응한다. 그러므로 정보 제공자들은 이러한 개인들의 성향을 만족시키기 위해서는 획일화된 정보보다는 소비자들이 스스로 판단할 수 있도록 다양한 정보를 제공해 주어야만 할 것이다. 이를 위하여 비용의 비교를 통해 경로를 선택하는 기존 알고리즘과 달리 최저비용과의 차이를 통한 알고리즘을 제안한다. 이를 위해 본 연구는 기존의 노드 기반 탐색법에 비해 네트워크 구조의 변화 없이 효율적으로 환승이나 회전제약을 표현할 수 있는 링크 기반 탐색법을 기반으로 운전자들의 다양한 needs를 최대한 반영할 수 있는, 즉 유연한 탐색 알고리즘의 개발을 목표로 한다. 이러한 목표를 위해, 기존의 최적 경로와 다경로 탐색 알고리즘을 대상으로 이론적 배경을 고찰하고, 다목적 정보 제공을 위한 다경로 탐색기법을 위한 동행원리를 개념화한 후, 이를 알고리즘에 적용하는 방안을 제안하며, 가상의 네트워크에 적용하여 알고리즘 수행과정을 보여주하고자 한다.

It is known that over one million car navigation devices are being currently used in Korea. Most, if not all, route guidance systems, however, provide only one "best" route to users, not providing any options for various types of users to select. The current practice dose not consider each individual's different preferences. These days, a vast amount of information became available due to the rapid development in information processing technology. Thus, users prefer choices to be given and like to select the one that suits him/her the best among available information. To provide such options in this paper, we developed an algorithm that provides alternative routes that may not the "least cost" ones, but ones that are close to the "least cost" routes for users to select.

The algorithm developed and introduced in the paper utilizes a link-based search method, rather than the traditional node-based search method. The link-based algorithm can still utilize the existing transportation network without any modifications, and yet enables to provide flexible route guidance to meet the various needs of users by allowing transfer to other modes and/or restricting left turns. The algorithm developed has been applied to a toy network and demonstrated successful implementation of the multi-path search algorithm for multi-purpose activities.

1. 서론

현재 우리나라의 경우 내비게이션 단말기 판매 대수는 60만대를 넘어설 것으로 전망되며, 이동통신사들의 텔레매틱스 서비스도 50만 명에 달할 것이라고 한다. 우리나라도 100만이 넘는 운전자가 경로탐색을 위한 단말기를 사용하고 있다는 것이다. 최근 연구 결과에서도, 현실성이 반영된 합리적인 경로를 효율적으로 산정할 수 있는, 최단 경로 탐색 알고리즘 개발의 중요성을 강조하고 있다(Kaufman and Smith, 1993). 그런데 지금까지 사업화된 길안내 모형들은 최적 경로 하나만을 제시함으로써, 이용자 스스로 판단할 수 있는 길을 원천적으로 막고 있다. 즉, 한 가지 정보만을 제공함으로써, 이용자가 자신의 경로를 선택할 적절한 비교대상을 제시해 주고 있지 못하는 것이다. 또한 최적 경로만을 제공하는 것은 네트워크상 최단 경로로 이용자들을 몰리게 하는 요소를 포함하고 있으므로, 오히려 지체를 유발시킬 수도 있는 것이다. 이러한 경로는 더 이상 최적 경로가 아니게 되어 버리는 것이다.

최적 경로 제시의 경우, 일반적으로 운전자의 경로 인지비용(Sheffi, 1985)의 확률적 속성 때문에 다양한 노선의 정보를 통해서 의사를 결정하도록 차선의 정보에 대한 제공영역을 넓히는 것이 필요하다고 한다(Rilett & Park, 1996). 그러므로 여러 대안 경로를 제공하는 다경로 알고리즘의 유연성은 최단 경로에 의존하는 것보다 현실적 정보제공 수단으로 활용가치가 매우 높다(이미영 외, 2003).

지금까지 대부분의 연구는 노드 기반의 최단 경로 탐색으로서, 이 경우 단일 교통망이 아닌, 복합 교통망을 표현하기 위해서는 환승을 위해 노드와 링크를 추가하여야만 표현이 가능했다. 또한, 현실적으로 존재하는 수많은 회전계약들에 의한 U-turn, P-turn 등을 직접적으로 표현할 수 없는 문제점 또한 포함하고 있다. 이러한 문제점들을 동시에 해결해 주는 것이 바로 링크 기반의 최단 경로 탐색 기법으로서, Potts & Oliver(1972)는 회전 제약 문제를 해결하기 위한 링크기반 경로 탐색법을 보여주었고, 노정현·남궁성(1995)은 링크기반 탐색 법을 Dijkstra 알고리즘에 적용한 경로 탐색법을 소개하였다. 환승을 고려한 링크 기반 탐색은 김현명 외(1999)에서 제시되었으며, 장인성(2000)은 환승 시간 제약을 고려한 링크 기반 탐색법을 제안하였다. 이에 본 논문에서는 링크 탐색법을 이용하여, 경로

를 탐색하였다.

다수경로 탐색 방법의 경우 대표적인 알고리즘으로는 Shier(1979)의 알고리즘과 Yen(1971)의 알고리즘이 있다. Martins(1984)와 Azevedo et al.(1993)은 Yen의 알고리즘을 루프경로까지 포함하여 탐색하는 방법을 제시하였고, 이미영 외(2003)는 Yen의 알고리즘을, 이미영 외(2004)는 Shier의 알고리즘을 이용하여 운전자들에게 차선의 정보에 대한 제공영역을 넓히기 위한 링크 기반 K 개의 다수 경로 탐색 알고리즘을 제시하였다. 본 논문에서는 Shier의 알고리즘과 Yen의 알고리즘을 비교한 후, 연구의 목적인 통행자의 다양한 needs를 반영할 수 있는 Yen의 알고리즘을 기반으로 탐색기법을 개발하였다.

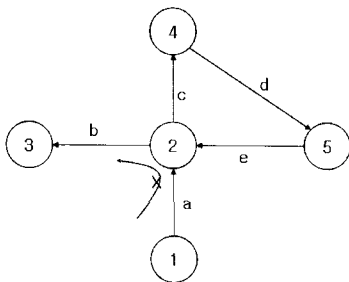
그러나 이러한 방법론들 모두는 아직까지 통행 목적 자체에만 비중을 두고 개발된 알고리즘으로서 다목적을 고려한 알고리즘은 오승(2004), Kang et al.(2006) 이외에 특별히 개발되고 연구된 것이 없는 실정이다. 두 연구에서 제시하는 알고리즘은 통행 목적이 결정된 후 휴리스틱한 방법으로 경로를 탐색하는 알고리즘이다. 하지만 통행자들은 자신들의 needs에 따라, 주어진 정보로부터 목적행동의 여부를 판단하므로 정보를 미리 제공한 후, 통행자 스스로 목적 행동을 결정할 수 있는 알고리즘을 제공해야 할 것이다. 그러므로 본 연구에서는 통행자들의 다양한 needs를 최대한 반영할 수 있는 알고리즘의 개발을 목표로 한다. 앞으로 이러한 알고리즘을 유연한 알고리즘(flexible search algorithm)이라 부르기로 한다. 여기서 말하는 다목적 다경로 탐색을 위한 유연한 알고리즘이라는 것은 집으로 가는 통행만 고려하는 것이 아니라, 도중에 서점에서 책을 사고, 식당에서 외식을 하며, 주유소에서 기름을 넣어서 돌아가려 하는 데, 3가지 목적 행동 각각의 선택 여부는 주어진 경로에 관한 정보를 바탕으로 이루어 질 수 있도록 하는 경로를 K 개 제공하는 것을 의미한다.

이러한 목표를 위해 기존의 최적 경로와 다경로 탐색 알고리즘을 대상으로 이론적 배경을 고찰하고, 다목적 다경로 탐색기법을 위한 통행원리를 개념화한 후, 링크 기반 탐색법을 이용하고, 기존의 다수경로탐색 기법을 응용하여, 최저가와의 차이라는 새로운 비용가치를 도입하여 사람들의 다양한 needs를 최대한 반영할 수 있는 새로운 알고리즘, 즉 유연한 탐색 알고리즘을 개발한 후, 가상의 네트워크에 적용하여 알고리즘 수행 과정을 보여주고자 한다.

II. 기존 알고리즘의 고찰

1. 링크 기반 탐색법

링크 기반의 최단경로탐색 알고리즘은 현 노드의 전 노드나 전전 노드를 검색해서 기점노드로부터 현 노드까지의 최단경로를 구축하는 노드기반의 순차적 탐색법과는 달리, 현 링크의 전 링크를 검색함으로써 기점노드로부터 현 링크의 종결노드까지의 최단경로를 구축한다(장인성, 2000). 일반적인 노드 기반 탐색법의 경우, 알고리즘의 특성상 노드의 중복을 허락하지 않으므로, 연속적인 회전제약이 발생하는 경우 적용하기가 힘들다. 하지만 링크 기반 탐색법의 경우 링크의 중복을 허락하지는 않지만, 노드의 중복을 허락하므로 더욱 현실에 근접한 탐색을 실시할 수 있다. 예를 들어 <그림 1>에서, 노드 기반 탐색 알고리즘의 경우는 노드1에서 노드3으로 가는 경로를 노드1-노드2-노드3으로 계산하지만, 노드2에서 노드3으로 가는 길은 좌회전 금지이므로, 최적경로는 노드1-노드2-노드4-노드5-노드2-노드3이 되어 노드의 중복이 발생된다. 이것을 링크로 표현하면 링크 a-링크 c-링크 d-링크 e-링크 b로서 링크의 중복 없이 회전제약을 고려할 수 있다.



<그림 1> 회전 제약이 있는 네트워크

이러한 링크 기반 탐색법은 최적 경로를 생각할 때, 회전제약 문제 때문에 U-turn이나 P-turn과 같이 노드의 중복은 무수히 많이 발생할 수 있지만, 같은 길을 다시 가는 경우, 즉 링크의 반복은 나타날 수 없는 현실에 더욱 맞는 탐색법이라 할 수 있다.

최단경로탐색 알고리즘은 출발 노드 r 에서부터 각 노드까지 비용을 계산하여 비교하여야 하므로, 링크 기반으로 탐색할 경우, 링크탐색 다음에 노드비용으로 전환해주는 단계를 거쳐야만 한다. 이는 해당링크로 들어

오는 최적 경로의 비용에, 해당링크의 값을 합하여 해당링크가 향하는 노드의 도착 비용으로 정하면 된다.

노드기반 탐색 방법이 출발지부터 탐색을 시작하여 먼저 탐색된 노드를 우선적인 다음 탐색노드로 고려하여 링크의 도착노드까지 비용을 계산한 후, 노드비용의 갱신이 더 이상 이루어지지 않을 만큼 반복하여 수행하는 반면, 링크기반 탐색 방법은 출발 노드에서 연결된 링크를 우선 탐색하여, 탐색된 링크를 다음 탐색링크로 선정하고, 선정된 링크의 도착 노드를 출발 노드로 하는 인접 링크의 링크비용을 갱신한다. 알고리즘은 링크비용의 갱신이 더 이상 이루어지지 않을 만큼 반복하여 수행한다. 알고리즘의 수행과정이 종료되면 노드표지에 근거한 비용으로 환산만 해주면 된다. 이 때 출발지 r 에서 링크 b 의 출발지점까지 최단 경로 비용을 계산하기 위해서는 두 개의 인접된 링크 a 와 b 로 구성된 표지를 이용한다. 이때 노드표지와와의 차이점은 두 인접링크에서 발생하는 회전비용(d_{ab})을 추가적인 네트워크의 확장 없이 고려할 수 있다는 것이다(이미영 외, 2003, 2004).

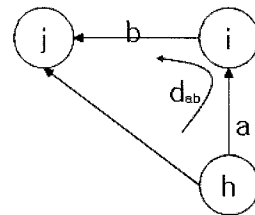
최단경로비용이 갱신되어 계산되는 수식은 다음과 같다(이미영 외, 2003).

$$\pi^{r_b} = \min_{a \neq b} \{ \pi^{r_a} + c_a + d_{ab}, \pi^{r_b} \}$$

π^{r_a} : 출발지 r 에서 링크 a 의 출발지점까지 최단 경로 비용

c_a : 링크 a 의 비용

d_{ab} : 링크 a, b 사이의 지체 비용



<그림 2> 링크기반 최단경로비용계산

2. 다경로 탐색 기법

다경로 탐색 기법은 출발지 r 에서 도착지 s 까지 최적 경로를 비용의 순서대로 K 개 제시하는 알고리즘으로서 크게 두 가지 방법이 있다. 하나는 기존의 표지 확정/교정의 방법을 확대 이용하는 것으로 K 개의 경

로를 동시에 탐색하는 방법(Shier, 1979)이고, 다른 하나는 최단경로알고리즘을 활용하여 탐색된(1~ K-1) 경로를 기반으로, 경로의 부분삭제를 통해 K 번째 다른 경로를 발견하는 방법이다(Yen, 1971).

Yen의 알고리즘이 최단경로로부터 다음 최단경로를 차례대로 도출해 넘으로써, 통행자의 needs에 맞는 정보를 유연하게 제공해 줄 수 있는 반면, Shier의 알고리즘은 지정된 조건에서 K 개의 경로를 동시에 탐색하므로, 제공되는 정보가 경직될 수밖에 없다. 예를 들어 통행자가 집으로 가는 도중에 A라는 행동과 B라는 행동을 하기를 원하는 데, B라는 행동은 지금 당장 할 필요는 없다면, Yen의 알고리즘은 B를 하였을 때와 하지 않았을 때의 경로를 유연하게 모두 제시할 수 있어서, 통행자에게 유연한 정보를 제공해 줄 수 있는 반면, Shier의 알고리즘은 B라는 행동을 할 지 말지를 미리 결정해야, 알고리즘에 반영할 수 있으므로, 결과에는 항상 했을 때, 또는 하지 않았을 때의 결과만 남게 된다.

즉 Yen의 알고리즘은 최단경로로부터 시작하고 K값을 미리 정할 필요가 없으므로, 최단경로에 행동 B를 할 수 있는 경로가 포함되어 있지 않다면, 행동 B를 할 수 있는 경로가 나올 때까지, 알고리즘을 반복수행하여, B를 하였을 때와 하지 않았을 때의 총 통행 비용 차이를 통행자에게 제시할 수 있다. 반면 Shier의 알고리즘은 행동여부와 K값 모두 미리 지정해 줘야 하므로, 다목적 다경로 탐색 알고리즘의 목적이 통행자에게 유연한 정보를 제공해 주는 것임을 고려할 때, Yen의 알고리즘이 더 적합하다고 볼 수 있다.

또한 Shier의 알고리즘은 K 개의 경로를 제시한 후에 사용자가 다음 경로를 원할 때 처음부터 다시 K+1 개를 계산하는 반면, Yen의 알고리즘은 이미 선택된 경로를 바탕으로 계산하므로 정보 제공 후 추가 정보 제공은 더 빠르다고 할 수 있다.

이에 본 연구는 Yen의 알고리즘을 이용한 이미영외(2003)의 방법론을 본 연구의 주제에 맞게 변형하여 적용한다.

III. 링크기반 다목적 다경로 탐색기법

1. 표식

r, s, i, j, k : 노드 지표, r, s 는 출발지와 도착지를 표시

- a, b : 링크 지표
- n : 구입 상품 지표
- c_a or c_{ij} : 연속된 노드 i, j 간 링크 a 의비용
- d_{ab} : 링크 a, b 사이의 지체비용
- m^n : 네트워크 내에 존재하는 상품 n 의 단위 가격 중 최저가
- b_a^n : 링크 a 에 존재하는 상품 n 의 단위 가격과 최저가(m^n)와의 차
- B^n : 상품 n 의 구입 수량
- C^n : 상품 n 의 구매 비용
- α^w : 통행자의 시간당 단위 비용
- t_a^k : 수단 k 를 이용하여 링크 a 를 통과하는 시간
- C^w : 통행자의 시간 비용
- $P^{kr,s}$: 출발지 r 과 도착지 s 를 연결하는 경로 중 수단 k 를 이용한 경로
- d_a^k : 수단 k 의 거리와 관련된 비용들의 단위 비용
- d_a^k : 수단 k 를 통해 경로 a 를 이동한 거리
- C^d : 통행자의 거리 관련 비용
- $G_A^{r,s}$: 출발지 r 에서 도착지 s 사이에서 최적으로 선택된 b_a^n 집합
- $G_B^{r,s}$: 출발지 r 에서 도착지 s 사이에 존재하는 모든 b_a^n 집합
- $K_A^{r,s}$: 출발지 r 과 도착지 s 를 연결하는 선택된 경로집합
- $K_B^{r,s}$: 출발지 r 과 도착지 s 를 연결하는 경로 중 선택되기 위해 대기하고 있는 경로집합
- $K^{r,i}$: $P_k^{r,s}$ 를 결정하기 위해 사용된 $R_k^{r,i}$ 의 경로집합
- $P_K^{r,s}$: 출발지 r 과 도착지 s 를 연결하는 선택된 경로 중 K 번째 최단경로.
- $R_K^{r,i}$: $P_{K-1}^{r,s}$ 에서 출발지 r 과 경유지 i 를 연결하는 경로
- S_j^s : 노드 j 에서 도착지 s 까지의 최단 경로
- $C[*]$: 경로 *의 비용

2. 비용계산

비용 계산과 관련되어서는 김창호(2003)에서 소개

된 방법을 기반으로 상품구매 가격 대신 최저가와와의 차이를 사용하며, 몇 가지 개념을 단순화 한다.

다목적 다경로 알고리즘에서 "현재 나의 위치에서 출발하여 10리터의 휘발유를 넣기 위해 주유소를 들르며, 책 두 권을 사고, 케이크 하나를 사서 집으로 가는 최소 비용의 경로를 찾아라."라는 여러 개의 통행목적의 존재한다면, 그 때의 비용 계산은 다음과 같다.

1) 구입비용

위의 예제에서 구입해야 할 상품은 세 개다. 10리터의 휘발유를 구입하는 행동은 $n=1, B^1=10$, 두 권의 책을 사는 행동은 $n=2, B^2=2$, 케이크 하나를 구입하는 행동은 $n=3, B^3=1$ 로서 표현되고, 링크 a 에 존재하는 상품 n 의 단위가격과 최저가(m^n)와의 차이는 b_a^n 으로 표현된다.

이 중 선택된 최적 경로에 여러 개의 상품 구입 장소가 존재한다면, $\min\{b_a^n\}$ 인 링크에서 구매가 이루어진다. 현실에 맞게 설명하기 위해서는 각각의 정차 비용 또한 고려해야 하나, 상품 구매 링크 선택 알고리즘이 무척 복잡해지므로, 본 연구에서는 제외한다.

구입비용에서 각 상품의 최저가와와의 차이를 적용하는 것은 통행자들에게 유연한 정보를 제공하기 위하여 서이다. 즉, 통행자가 귀가 도중 상품 1과 상품 2를 구입하고자 하는데, 만약 둘 다 모두 당장 구입해야 한다면, 두 상품의 가격을 모두 고려하여 비교가 가능하지만, 크게 급한 것이 아니라면, 여러 가지 상황을 고려할 수 있도록 해야 하는 것이다.

즉, <표 1>과 같은 경로들을 생각해보자

<표 1> 상품비용으로 비교한 경로선택의 예

	통행비용	상품 1 비용	상품 2 비용	총 비용
경로 1	200	5000		5200
경로 2	200		2000	2200
경로 3	180			180
...
최저가		5000	1500	

이 때, 총 통행비용으로만 최적 경로를 도출하면, 경로 3-경로 2-경로 1이 될 것이다. 하지만, 상품 1과 2의 가격 차이에 의해, 경로 2의 경우는 최저가 보다 500원이나 더 비싸게 주고 사야 하는 경우이지만, 상품 1의 경우는 최저가로 구입할 수 있는 것이다. 만약

상품 1의 평균가격이 10000원이고 경로 1에서 반짝 대박세일을 하고 있다면, 통행자는 경로 2를 포기하고 경로 1을 선택할 것이다. 그러나 총통행비용으로 K 개를 정렬했을 경우 고가품일수록 할인 폭이 무척 크다고 할지라도 다른 상품과의 가격차이로 인해, K 개에 선택되지 않을 확률이 무척 높다.

이를 방지하기 위해 비용을 최저가와와의 차이로 계산하면 결과는 <표 2>와 같다.

<표 2> 최저가로 비교한 경로 선택의 예

	통행비용	상품 1 비용	상품 2 비용	총 비용
경로 1	200	0		200
경로 2	200		500	700
경로 3	180			180
...
최저가		5000	1500	

이 경우, 상품 2가 최저가 보다 무척 비싸므로 그 결과가 총 통행비용에 그대로 반영될 수 있다. 이러한 경로가 만약 K 개에 포함된다면, 구입하지 않았을 때와 비용차가 무척 크므로, 통행자는 상품 구매를 미룰지, 말지를 충분한 정보로 판단할 수 있을 것이다.

이 때 $C^n = B^n \times \{\min\{b_a^n\} + m^n\}$ 으로 계산된다.

2) 시간 비용

각 통행자의 시간당 단위 시간비용을 알고 있다고 가정하고 그것을 α^w 라고 표기한다. 이 때 출발지 r 에서 도착지 s 까지 총 이동시간을 $\sum_a t_a^k$ 라 하면, 통행자의 시간 비용은 다음과 같다.

$$C^w = \alpha^w \sum_k \sum_{a \in P^{ks}} t_a^k$$

3) 거리 관련 비용들

연료 소비, 차량 감가상각비, 대중교통 요금처럼 거리와 관련된 비용들의 Km당 단위 비용을 d^k 라 하고, 수단 k 를 통해 이동한 총 거리를 d_a^k 라고 하면, 총 거리 관련 비용 C^d 는 다음과 같다.

$$C^d = \sum_k d^k \sum_{a \in P^{ks}} d_a^k$$

그러므로 출발지 r 에서 도착지 s 까지 통행 비용은 다음과 같이 정의할 수 있다.

$$C[P^{rs}] = C^w + C^d$$

통행 비용을 기준으로 통행 경로를 도출하고, 상품 구매 비용은 통행자의 기호에 따라 최종 결과에 전부 반영하거나 일부 반영하여 통행자의 needs에 맞는 K 개의 최적 경로를 도출한다.

이러한 비용을 바탕으로 한 링크기반 다목적 다경로 탐색기법 알고리즘은 다음과 같다.

3. 알고리즘

본 연구는 Yen의 알고리즘을 이용한 이미영 외 (2003)의 방법론을 기반으로 몇 가지 방법을 추가하여 새롭게 개발한 알고리즘으로, 첫 번째 단계인 최적경로 P_1^{rs} 탐색과정에서 역방향 링크 기반 최단 경로 탐색 방법을 사용하여 각 노드에서 도착지 s 까지의 최단경로를 찾는다. 이 때 저장된 각 노드와 도착지까지의 최단경로를 K 번째 경로를 탐색할 때 사용하므로, 각 단계마다의 경로 탐색 과정이 생략되는 새로운 알고리즘이다. 또한 P_k^{rs} 를 결정하기 위해 사용된 R_k^{rs} 의 경로집합 K^{rs} 를 새롭게 도입하여, 이전에 탐색된 경로와의 반복탐색을 피하게 함으로써 알고리즘의 효율을 높였다. K 번째 경로를 선택하기 위해서는 저장된 K_B^{rs} 중 가장 작은 값과 새롭게 추가된 값들을 비교하면 되므로 상당히 효율적인 경로 탐색이 가능하다. 각 단계는 다음과 같다

【step 1】 P_1^{rs} , S^{js} 결정

1) P_1^{rs} , S^{js} 결정

- (1) 역방향 링크 기반 최단경로 탐색 방법을 사용하여 도착지 s 와 모든 노드간의 최단 경로를 찾은 후, 출발지 r 에서 도착지 s 까지의 최단경로를 P_1^{rs} 라 한다. 이 때, 각 경로마다 통행비용을 $C[S^{js}]$ 라 한다.

- (2) 경로를 구성하는 링크에 b_a^i 가 존재하면, G_B^{js} 에 추가한다.

$$G_B^{js} = G_B^{js} \cup \{b_a^i\}$$

- (3) 선택된 경로를 K_A^{rs} 에 추가한다.

$$K_A^{rs} = K_A^{rs} \cup \{P_1^{rs}\}$$

【step 2】 ($k=2,3,\dots,K$) 반복진행으로 P_k^{rs} 부터 P_K^{rs} 의 결정

P_k^{rs} 를 발견하기 위하여 이미 P_1^{rs} , P_2^{rs} , ..., P_{k-1}^{rs} 가 이미 K_A^{rs} 에 포함되어 있음.

1) 탐색금지 링크를 통한 R_K^{rs} 선정

- (1) P_{k-1}^{rs} 경로에서 도착지 s 하나 전 링크부터 차례대로 탐색금지 링크를 부여하면서, R_K^{rs} 를 구성한다.

즉, P_{k-1}^{rs} 가 (r)-(2)-(3)-(4)-(s) 라면,

$$R_K^{rs} = \{(r)-(2)-(3)-(4)\},$$

$$R_K^{rs} = \{(r)-(2)-(3)\},$$

$$R_K^{rs} = \{(r)-(2)\}$$

- (2) 이 때, R_K^{rs} 에서 노드 연결 순서가 (r)-(2)---(i-1)-(i)이고, $R_K^{rs} \in K^{rs}$ 이면, R_K^{rs} , $R_K^{(i-1)}$, ..., $R_K^{(2)}$ 는 이미 그 전 단계에서 계산이 되었으므로, 고려할 필요가 없다.

- (3) $R_K^{rs} \notin K^{rs}$ 이라면, $K^{rs} = K^{rs} \cup \{R_K^{rs}\}$ 을 한 후, 다음 단계로 넘어간다.

- (4) 상품 집합은 다음과 같이 된다.

$$G_B^{rs} = G_B^{rs(i+1)} - G_B^{rs(i+1)}$$

2) K_B^{rs} 의 구성

선정된 R_K^{rs} 에서 i 번째 노드와 연결된 노드 중 P_{k-1}^{rs} 에서 연결되었던 노드를 제외한 다른 모든 노드들을 선택한 후, 이미 계산된 S^{js} 에서 선택된 노드들로부터 도착지 s 까지 최단경로를 첨가하여, 대기 경로 집합 K_B^{rs} 에 추가한다.

즉, $R_K^{rs} = \{(r)-(2)-(3)-(4)\}$ 에서,

노드 (4)에서 연결된 노드가 (5), (6), (s) 이고,

$S^{js} = \{(5)-(7)-(s)\}$, $S^{js} = \{(6)-(8)-(9)-(s)\}$ 라면,

대기 경로 집합 K_B^{rs} 에 추가되는 경로는

$$R_K^{rs} + S^{js} = \{(r)-(2)-(3)-(4)-(5)-(7)-(s)\}$$

$$R_K^{rs} + S^{js} = \{(r)-(2)-(3)-(4)-(6)-(8)-(9)-(s)\}$$

가 된다.

이 때 반드시 구입해야 하는 상품이 n 이고, 선택된 경로가 링크a - 노드i - 링크b - 노드j - 링크c 와 같다면(—(a)—(i)—(b)—(j)—(c)—), 비용과 상품

집합은 다음과 같다.

비용 : $C[R_K^{rs}] + d_{ab} + c_b + d_{bc} + C[S^{js}] + \min\{b_a^n\}$

상품 집합 : $G_B^{rs} = G_B^{rs} \cup G_B^{ij} \cup G_B^{js}$
 $(b_a^n \in G_B^{rs})$

3) K번째 경로의 선정

K_B^{rs} 에서 가장 작은 비용을 가진 경로를 K번째 경로 P_K^{rs} 로 선정하여 K_A^{rs} 에 포함시키고, K_B^{rs} 에서 제외한다.

$P_K^{rs} = \min_{(p)} C[p]$, $\forall p \in P_B^{rs}$

$K_A^{rs} = K_A^{rs} \cup \{P_K^{rs}\}$, $K_B^{rs} = K_B^{rs} - \{P_K^{rs}\}$

4) K번째 경로에서 상품 구입비용 계산.

G_A^{rs} 는 G_B^{rs} 의 b_a^n 중 최소값들로 구성되고, $\min\{b_a^n\}$ 에 m^n 을 더하여, 원래 가격으로 변환한 후, 구입 수량 B^n 을 곱하여 상품 구입비용 C^n 을 구한다.

$G_A^{rs} = \min\{b_a^n\}$ ($b_a^n \in G_B^{rs}$)

$C^n = B^n \times (\min\{b_a^n\} + m^n)$

5) 사용자가 선택한 조건에 맞는 결과가 제시되면 알고리즘 종료

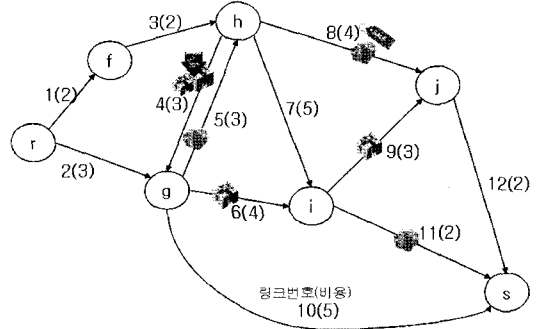
- (1) 상품을 구매하지 않고, 통행만 하는 경우, 통행 비용만으로 K 개의 최적 경로를 제공한다.
- (2) 상품 n을 반드시 구매해야 한다면, 구매하지 않았을 때와의 비교를 위하여 한 두 개 정도의 최적통행경로를 제시한 후, b_a^n 을 고려한 통행 가격으로 정렬된 K 개를 제공한다.
- (3) 상품 n을 반드시 구매할 필요가 없다면, 순차적으로 계속 K 개를 제공할 때까지, G_B^{rs} 에서 b_a^n 이 존재하지 않는다면, b_a^n 이 있는 경로가 나타날 때까지 경로를 제공하고 종료한다.

IV. 알고리즘의 적용

1. 실험 네트워크의 구성

(그림 3)과 같이 7개의 노드와 12개의 링크로 구성되어 있는 임의의 네트워크를 구성한다. 계산의 편의를 위하여 회전지체는 존재하지 않는다고 가정한다. 알파벳은 노드를 뜻하고, 화살표 위의 숫자는 링크를 뜻하

며, 괄호안의 숫자는 링크 비용을 뜻한다. 구입하고자 하는 상품은 2개이며, 각각 한 개씩 구매하기를 원한다. $B^1 = B^2 = 1$. 위치와 단위가격과의 차이는 <표 3>과 같다. 이 때, 각 상품의 단위가격 중 최저가는 $m^1 = 10$, $m^2 = 5$ 이다.



<그림 3> 실험 네트워크

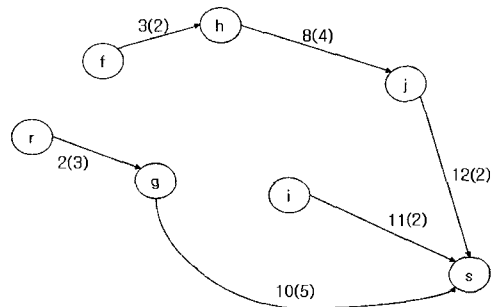
<표 3> 상품 구매 정보

상품	1	1	1	2	2	2
위치	4	9	6	5	11	8
가격차 (b_a^n)	(b_4^1)	(b_9^1)	(b_6^1)	(b_5^2)	(b_{11}^2)	(b_8^2)

2. 알고리즘의 적용

[Step 1] P_1^{rs} , S^{js} 의 결정

역방향 링크기반 최단 경로 탐색법을 사용하여 구한다.



<그림 4> 노드 s까지 최단 경로

<표 4> 노드 s까지 최단경로 정보

출발지	r	f	g	h	i	j
도착지	s	s	s	s	s	s
통행비용	8	8	5	6	2	2
G_B^{rs}	-	b_8^2	-	b_6^2	b_{11}^2	-

[Step 2] P_K^s 의 결정 - 통행비용만을 고려

앞에서 탐색된 $P_1^s(r-g-s)$ 를 기반으로 P_2^s 를 찾는다.

먼저 $g-s$ 를 연결하는 10번 링크를 탐색 금지 링크로 선정한다. 그러면 $R_2^g(r-g)$ 가 K^{ri} 에 저장되고, 노드 g 에서 새롭게 연결되는 링크를 찾으면 된다. 5번 링크($g-h$)와 6번 링크($g-i$) 두 가지가 다음 탐색 링크가 되고, 각각의 비용은 $R_2^g(r-g)$ 의 비용 - P_1^s 의 비용에서 탐색금지링크인 10번 링크의 비용을 빼면 된다. -에서 다음 탐색 대상인 5번 링크($g-h$)의 비용을 더하

고, 미리 저장된 S^{hs} 의 비용을 더하면 경로의 비용이 결정된다. 6번 링크의 경우에도 동일한 방법으로 결정하면 된다. 노드 g 에 더 이상 연결된 링크가 없으므로, 그 다음 탐색금지링크를 선정하도록 한다. 2번 링크($r-g$)가 탐색 금지 링크가 되고, $R_2^r(r)$ 을 K^{ri} 에 저장한 후 노드 r 에서 연결된 1번 링크($r-f$)가 다음 탐색 링크가 된다. 이 때 비용 계산 역시, 앞서 계산했던, $R_2^g(r-g)$ 의 비용에서 탐색금지링크인 2번 링크의 비용을 뺀 후, 새로운 탐색링크인 1번 링크의 비용을 더하고, 미리 저장된, S^{fs} 의 비용을 더하면 된다. 더 이상의 탐색 금지링크가 없으므로 K_B^s 에 저장된 값 중 가장 작은 값을 P_2^s 로 결정하면 된다. 나머지 단계도 이와 같은 방법을 실시하면, 그 결과는 <표 5>와 같다.

통행 비용의 순으로 링크의 반복 없이 결과가 나온 것을 볼 수 있다.

<표 5> 통행비용만 고려한 다경로 탐색 결과

K	K^{ri}	K_B^s	$C(R_K^r) + c_g + C(S^s)$	K_A^s	$C(P^s)$
1				rg-s	8
2	rg r	rg-hjs rg-is r-f-hjs	3+3+6=12 (5) ¹⁾ 3+4+2=9 (2) 0+2+8=10 (3)	rg-is	9
3	rg-i	rg-ij-s rg(X) ²⁾	7+3+2=12 (6)	r-f-hjs	10
4	r-f-hj r-fh r-f	r-f-hj(X) ³⁾ r-f-h-g-s r-f-h-i-s r-f(X)	4+3+5=12 (7) 4+5+2=11 (4)	r-f-h-i-s	11
5	r-f-h-i	r-f-h-i-j-s r-f-h(X)	9+3+2=14 (10)	r-g-h-j-s	12
6	rg-hj rg-h	rg-hj(X) rg-h-g-s rg-h-i-s rg(X)	6+3+5=14 (11) 6+5+2=13 (8)	rg-ij-s	12
7	rg-ij	rg-ij(X) rg-i(X)		r-f-h-g-s	12
8	r-f-h-g	r-f-h-g-hjs r-f-h-g-is r-f-h(X)	7+3+6=16 (13) 7+4+2=13 (9)	r-g-h-i-s	13
9	rg-h-i	rg-h-i-j-s rg-h(X)	11+3+2=16 (14)	r-f-h-g-i-s	13
10	r-f-h-g-i	r-f-h-g-i-j-s r-f-h-g(X)	11+3+2=16 (15)	r-f-h-i-j-s	14
11	r-f-h-i-j	r-f-h-i-j(X) r-f-h-i(X)		rg-h-g-s	14
12	rg-h-g	rg-h-g-is rg-h(X)	9+4=2=15 (12)	rg-h-g-is	15
13	rg-h-g-i	rg-h-g-i-j-s rg-h-g(X)	13+3+2=18 (17)	r-f-h-g-hjs	16
14	r-f-h-g-hj r-f-h-g-h	r-f-h-g-hj(X) r-f-h-g-h-i-s r-f-h-g(X)	10+5+2=17 (16)	rg-h-i-j-s	16
15	rg-h-i-j	rg-h-i-j(X) rg-h-i(X)		r-f-h-g-i-j-s	16
16	r-f-h-g-i-j	r-f-h-g-i-j(X) r-f-h-g-i(X)		r-f-h-g-h-i-s	17
17	r-f-h-g-h-i	r-f-h-g-h-i-j-s r-f-h-g-h(X)	15+3+2=20 (18)	rg-h-g-i-j-s	18
18	rg-h-g-i-j	rg-h-g-i-j(X) rg-h-g-i(X)		r-f-h-g-h-i-j-s	20
19	r-f-h-g-h-i-j	r-f-h-g-h-i-j(X) r-f-h-g-h-i(X)			

3. 알고리즘 결과

본 연구의 목적이 통행자의 needs를 고려한 유연한 정보 제공을 위한 다목적 다경로 탐색 알고리즘의 개발이므로, 가능한 요구 4가지에 따라 각각의 결과를 보면 다음과 같이 나타난다. 통행자의 요구가 $K=5$ 라고 가정하고, 최단경로와의 비교를 위하여 최단경로를 기본으로 제공받기를 원한다면, $K=6$ 일 때까지 알고리즘이 반복된다.

1) 통행자가 상품 1과 2 모두 반드시 구입하고자 하는 경우

<표 6> 상품 1, 2 모두 존재하는 경로들

K	최적경로 P_K^s	$C(P^s)$	G_B^s	min ν	min ν	총 비용	실제 비용	구매 상품
1	rg-s 2-10	8		-	-	8	8	-
2	rg-is 2-6-11	9	b_1, b_1	1	2	12	27	1, 2
3	r-f-h-g-i-s 1-3-4-6-11	13	b_1, b_1, b_1	0	2	15	30	1, 2
4	rg-h-g-s 2-5-4-10	14	b_1, b_1	0	1	15	30	1, 2
5	rg-h-g-i-s 2-5-4-6-11	15	b_1, b_1, b_1	0	1	16	31	1, 2
6	r-f-h-g-hjs 1-3-4-5-8-12	16	b_1, b_1, b_1	0	0	16	31	1, 2

- 1) 괄호 안의 숫자는 K값
- 2) K^{ri} 에 이미 앞에서 탐색된 경로가 있으므로 중단
- 3) 탐색 금지 링크 이외에 새롭게 연결되는 링크가 없으므로 중단

상품 1과 상품 2가 동시에 존재하는 경로들만 고려하여 [통행비용 + 상품 1의 비용차 + 상품 2의 비용차] 순서로 정리하고, 통행시간의 비교를 위하여, 최단경로는 기본으로 제공한다. 비용에 무척 민감한 통행자의 경우 최단 경로와 상품 구입 시 경로와의 차이가 무척 크다면, 구입통행을 포기할 수도 있으므로, 최단경로를 제시해 줄 필요가 있다.

2) 통행자가 상품 1은 반드시 구입, 상품 2는 당장 구입할 필요가 없는 경우

상품 1이 반드시 존재하는 경로만 고려하여 [통행비용 + 상품 1의 비용차] 순서로 정리하고, 통행시간의 비교를 위하여, 최단경로는 기본으로 제공한다. 비용에 무척 민감한 통행자의 경우 최단 경로와 상품 구입 시 경로와의 차이가 무척 크다면, 구입통행을 포기할 수도 있으므로, 최단경로를 제시해 줄 필요가 있다.

<표 7>과 같은 정보가 제공될 경우, 상품 2의 구매여부는 상품 2에 대한 통행자의 민감도에 따라 결정된다. 통행자가 상품 2의 가격에 민감하다면, 구매를 포기하고 통행할 수 있는 것이다. 이러한 판단을 제공하기 위해서는 최적경로 선정에 상품 2의 가격이 포함되어서는 안 된다.

<표 7> 상품 1이 존재하는 경로들

K	최적경로 P_K^*	$C(P^*)$	G_B^*	min b^*	min b^*	총 비용	실제 비용	구매 상품
1	r-g-s 2-10	8		-	-	8	8	-
2	r-g-i-s 2-6-11	9	b_6, b_{11}	1	2	10	27	1, 2
3	r-f-h-g-s 1-3-4-10	12	b_4	0		12	22	1
4	r-g-i-j-s 2-6-9-12	12	b_6, b_9	1		13	23	1
5	r-f-h-g-i-s 1-3-4-6-11	13	b_4, b_6, b_{11}	0	2	13	30	1, 2
6	r-g-h-g-s 2-5-4-10	14	b_5, b_4	0	1	14	30	1, 2

3) 통행자가 상품 2는 반드시 구입, 상품 1은 당장 구입할 필요가 없는 경우

상품 2가 반드시 존재하는 경로만 고려하여 [통행비용 + 상품 2의 비용차] 순서로 정리하고, 통행시간의 비교를 위하여, 최단경로는 기본으로 제공한다. 비용에 무척 민감한 통행자의 경우 최단 경로와 상품 구입 시 경로와의 차이가 무척 크다면, 구입통행을 포기할 수도

있으므로, 최단경로를 제시해 줄 필요가 있다.

<표 8>과 같은 정보가 제공될 경우, 상품 1의 구매여부는 상품 1에 대한 통행자의 민감도에 따라 결정된다. 통행자가 상품 1의 가격에 민감하다면, 구매를 포기하고 통행할 수 있는 것이다. 이러한 판단을 제공하기 위해서는 최적경로 선정에 상품 1의 가격이 포함되어서는 안 된다.

<표 8> 상품 2가 존재하는 경로들

K	최적경로 P_K^*	$C(P^*)$	G_B^*	min b^*	min b^*	총 비용	실제 비용	구매 상품
1	r-g-s 2-10	8		-	-	8	8	-
2	r-f-h-j-s 1-3-8-12	10	b_8		0	10	15	2
3	r-g-i-s 2-6-11	9	b_6, b_{11}	1	2	11	27	1, 2
4	r-g-h-j-s 2-5-8-12	12	b_5, b_8		0	12	17	2
5	r-f-h-i-s 1-3-7-11	11	b_{11}		2	13	18	2
6	r-g-h-i-s 2-5-7-11	13	b_5, b_{11}		1	14	19	2

4) 통행자가 상품 1, 2 모두 당장 구입할 필요가 없는 경우

<표 9> 모든 경로들

K	최적경로 P_K^*	$C(P^*)$	G_B^*	min b^*	min b^*	총 비용	실제 비용	구매 상품
1	r-g-s 2-10	8		-	-	8	8	-
2	r-f-h-j-s 1-3-8-12	10	b_8	-	0	10	15	2
3	r-g-i-s 2-6-11	9	b_6, b_{11}	1	2	12	27	1, 2
4	r-g-h-j-s 2-5-8-12	12	b_5, b_8	-	0	12	17	2
5	r-f-h-g-s 1-3-4-10	12	b_4	0	-	12	22	1
6	r-f-h-i-s 1-3-7-11	11	b_{11}	-	2	13	18	2

모든 경로에 대하여 [통행비용+각 상품의 비용차 (없을 경우는 0으로 처리)] 순서로 정리하고, 통행시간의 비교를 위하여, 최단경로는 기본으로 제공한다. 비용에 무척 민감한 통행자의 경우 최단 경로와 상품 구입 시 경로와의 차이가 무척 크다면, 구입통행을 포기할 수도 있으므로, 최단경로를 제시해 줄 필요가 있다.

통행비용만을 고려하였을 때의 결과와 다른 순서로 정렬되어 제시. 이 경우는 통행자가 최단 경로와의 비용 차이, 또는 최저가와와의 비용 차이를 스스로 판단하

여, 통행 이외의 목적통행을 실시 할지 아닐 지 자신의 가치에 따라 결정할 수 있게 된다.

첫 번째의 경우와 차이가 나는 것은 비용 계산 방법은 똑같지만, 정렬할 때 첫 번째 방법은 상품 1과 상품 2가 동시에 존재하지 않는 경로는 제외되는 반면, 이 경우는 상품의 포함여부에 상관없이 전체 비용으로 정렬하기 때문이다.

V. 결론

IT 산업이 발달하고, 정보의 양이 넘쳐날수록, 사람들은 확립화되어 제공되는 정보 보다는, 스스로 다양한 방법을 통해 정보를 찾아내며, 이를 가공하여 판단하고, 반응한다. 그러므로 정보 제공자들은 이러한 개인들의 다양한 needs를 만족시키기 위해서는 확립화된 정보보다는 소비자들이 스스로 판단할 수 있도록 다양하고 유연한 정보를 제공해 주어야만 할 것이다.

하지만 지금까지 제시되어 온 대부분의 경로 탐색 알고리즘은 최단 경로에만 국한되거나, 통행자체만을 고려하였다. 다목적에 고려하고 있더라도, 반드시 그 목적통행이 일어나야 한다고 가정함으로써, 더욱 다양해 질 수 있는 통행자의 판단을 배려하지 못하고 있다. 이에 본 연구는 지금까지 개발된 탐색 알고리즘 기법을 응용하여 유연한 정보 제공을 위한 다목적 다경로 탐색 기법을 개발하였다.

일반적으로 최적 경로를 찾기 위해 시간비용이라는 개념을 사용하는 데, 사람들이 느끼는 시간 가치라는 것은 상황에 따라서 매우 달라질 수 있는 것이다. 그러므로 사람들의 다양한 needs를 반영할 수 있는, 즉, 통행자가 본인의 상황을 고려하여, 적절한 판단을 내릴 수 있도록 유연한 정보를 제공하는 본 알고리즘은 충분히 유의미하다.

추후 연구과제로는 다수단을 고려한 알고리즘의 개발과 정차비용을 고려한 알고리즘의 개발이다.

대중교통의 발달에 의하여 다수단을 고려한 알고리즘의 개발이 필요한데, 이는 각 수단별로 링크를 부여함으로써 어느 정도 해결할 수 있는 문제이다.

이와는 별도로 정차비용을 고려한 알고리즘의 개발이 필요하다. 본 연구에서 여러 목적 통행 시 발생하는 정차 비용을 제외하였는데, 이는 현실적으로 문제가 될 수 있는 부분이다. 예를 들어 링크 1에서 상품 1의 구입비가 20, 정차비가 30, 링크 2에서 상품 1의 구입

비가 40, 상품 2의 구입비가 20 정차비가 20 인데, 경로 2에서는 한 번의 정차로 상품1과 상품 2를 동시에 구입할 수 있다고 가정하자.

〈표 10〉 링크별 상품 구입비 및 정차비

	상품1	상품2	정차비
링크1	20		30
링크2	40	20	20

이 때, 상품 1과 상품 2를 반드시 구입해야 한다면, 최저가만을 택하는 경우 링크 1에서 상품 1을 링크 2에서 상품 2를 구입하는 것이 최적이나, 정차비용을 포함하면, 링크 1과 링크 2에서 각각 구입하는 비용은 $20+30+20+20=90$ 이 된다. 하지만, 링크 2에서 상품 1, 상품2를 동시에 구입하면, $40+20+20=80$ 으로, 이것이 최적 경로가 되어야만 한다. 이러한 정차 문제를 해결하기 위해서는 경로 내에 존재하는 모든 상품 구입지의 가격과 정차비용을 모두 고려하여 조합의 문제로 풀어야만 하므로 매번 경로 선택마다, 이러한 비교를 하는 것은 연산수의 기하급수적인 증가를 가져온다. 그러므로 이러한 정차 비용을 고려한 알고리즘의 개발이 이루어져야, 현실에 더욱 가까운 알고리즘이 될 것이다.

참고문헌

1. 강맹규(1991), "네트워크와 알고리즘", 박영사.
2. 김창호(2003), "위치기반서비스를 위한 복합 교통수단의 통행경로 모형", 한국LBS학회논문지 창간호, 제1권 제1호, 한국LBS학회, pp55~63.
3. 김현명·임용택·이승재(1999), "통합교통망 수단 선택-통행배정모형 개발에 관한 연구", 대한교통학회지, 제17권 제5호, 대한교통학회, pp87~98.
4. 노정현·남궁성(1995), "도시가로망에 적합한 최단 경로탐색 기법의 개발", 대한국도도시계획학회지, 제30권 제5호, 대한국도도시계획학회, pp153~168.
5. 신성일(2004), "교통망에 적합한 K 비루프 경로 탐색 알고리즘", 대한교통학회지, 제22권 제6호, 대한교통학회, pp121~131.
6. 오승(2004), "위치기반서비스를 위한 휴리스틱 경로탐색 모형의 개발", 서울대학교 박사학위논문.
7. 이미영·유기운·김정현·신성일 (2003), "덩굴당 통행 패턴을 고려한 One-To-One 다경로 알고리즘", 대한교통학회지, 제21권 제6호, 대한교통학회, pp8

- 9~99
8. 이미영 · 백남철 · 최대순 · 신성일 (2004), "링크폴 지깅신 다수경로탐색 알고리즘", 대한교통학회지, 제22권 제2호, 대한교통학회, pp131~143.
 9. 임용택 (2004), "일반가로망에서 교통정보 제공을 위한 n-path 알고리즘의 개발", 대한교통학회지, 제22권 제4호, pp135~144.
 10. 장인성 (2000), "서비스 시간 제약이 존재하는 도시부 복합 교통망을 위한 링크 기반의 최단경로탐색 알고리즘", 대한교통학회지, 제18권 제6호, 대한교통학회, pp 111~121.
 11. Azevedo, J. A. , M. E. O. S. Costa , J. J. E. R. S. Madeira , and E. Q. V. Martins (1993), "An algorithm from the Ranking of Shortest Paths", European Journal of Operational Research, Vol. 69, pp97~106.
 12. Kaufman, D. E. and R. L. Smith (1993), "Faster Paths in Time-Dependent Networks for IVHS Applications", IVHS Journal. Vol. 1. pp1~11.
 13. Kim, T. J. (2004), "Multi-modal routing and navigation cost functions for location-based services (LBS). In Urban and Regional Transportation Modeling: Essays in Honor of David Boyce (D. H. Lee ed.)", Edward Elgar, Northampton, MA.
 14. Martins, E. Q. V. (1984), "An Algorithm for Ranking Paths that May Contain Cycles", European Journal of Operational Research, Vol. 18, pp123~130.
 15. Potts, R. B. , R. M. Oliver (1972), "Flows in transportation networks", Academic press.
 16. Rilett, L. R. and D. Park (1996), "Identifying Unique Routes in Transportation Networks". Presented at the 3rd Annual World Congress on Intelligent Transportation Systems, Orlando Florida, October.
 17. Kang, S. and T. J. Kim (2005), "Solving A Location-Based Concierge Service Problem: A Heuristic Approach Using Genetic Algorithm", Presented at the Symposium on Societies and Cities in the Age of Instant Access, November 2005, University of Utah, Salt Lake City, UT.
 18. Kang, S. , S. Oh and T. J. Kim (2006), "A Heuristic Algorithm for Solving a Multimodal Location-Based Concierge Service Problem", Presented at the 85th Annual Meeting of the Transportation Research Board, Washington D.C.
 19. Sheffi, Y. (1985), "Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods", Englewood Cliffs, N.J.: Prentice-Hall.
 20. Shier, R. D. (1979), "On Algorithms from Finding the k Shortest Paths in a network", Networks. Vol.9, pp.195~214.
 21. Yen, J. Y. (1971), "Finding the K shortest Loopless Paths in a network", Management Science. Vol.17, pp711~715.

✉ 주 작 성 자 : 정연정

✉ 교 신 저 자 : 정연정

✉ 논문투고일 : 2006. 2. 25

✉ 논문심사일 : 2006. 4. 21 (1차)

2006. 4. 24 (2차)

2006. 4. 25 (3차)

✉ 심사판정일 : 2006. 4. 25

✉ 반론접수기한 : 2006. 9. 30