一般論文

다수개의 여행시간이 주어진 경우의 지정된 마디간의 최단경로 문제

이명석* · 박순달*

A Study on Shortest Problem Between Specified Nodes with Multiple Travel Time

Myung Seok Lee* · Soon Dal Pak*

Abstract

The purpose of this thesis is to find the shortest path between two nodes on an acyclic network where the arc costs are determined by the starting time at the starting node of the arc.

A branch and bound method for optimal solutions and a heuristic method is developed. In heuristic method Dijkstra algorithm is modified to maintain the minimum arrival times of maximum k informations in the each time period at each node and is updated by the result with the insertion technique.

Expermetal results among two methods are presented with regard to run time and solution qualities.

1. 서 론

최근 차량의 급속한 증가 도시로의 인구유입 등으로 인하여 도로의 사정은 점점 악화되어 왔다. 이같은 상황에서 두 지점의 최단경로를 찾는데 호상에 단순히 하나의 여행시간이 주어진 경우를 고려하는 것은 적합치 않다. 그러므로 호상에 시간에 따라 영향받는 다수개의 여행시간을 고려한 두 지

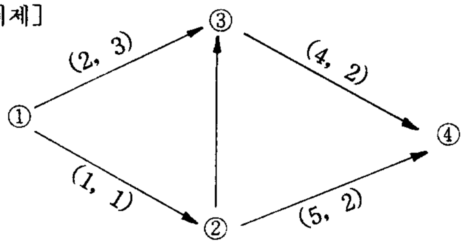
* 서울대학교 산업공학과

점간의 최단경로를 찾는것이 적합하다.

본 논문에서는 무환네트워크 상에서 각 호에는 시간대에 따라 영향을 받는 k개의 여행시간이 주어지고 출발마디, 목적마디와 출발마디에서의 출발시각이 주어질때 출발마디와 목적마디간의 최단경로를 구하는 문제를 다룬다. 이러한 형태의 문제는 Call Taxi회사, 회사의 운송부 등에서 적용할 수 있을 것이다. 문제에 주어진 가정은 마디의 도착시각이 임의시간대에 들어가면 그 마디에서 출발하는 호는 그 마디의 시간대에 영향을 받는다. 예를들면, 아래 그림에서 경로 1-2-4을 거치는 경우 마디 2로의 도착시각은 8시 28분이다. 이때 경로 2-4을 지날때의 여행시간은 마디 2의 도착시각이 8시 30분전이므로 여행시간은 5분을 적용한다.

이러한 문제는 호상에 단 하나의 여행시간이 주어진 경우처럼 최단경로의 부분경로는 최단경로라 보장할 수 없다. 즉, 예제에서 보면

[예제]



출발시각 : 8 : 27

출발마디 : ①　　목적마디 : ④

(t_1, t_2) :

t_1 : 8 : 30전의 여행시간

t_2 : 8 : 30분후의 여행시간

[그림 1-1]

가능한 경로	1-3-4	경로비용 6
	1-2-3-4	경로비용 5
	1-2-4	경로비용 6

여기서 최적경로는 1-2-3-4이다. 이때 1에서 3까지의 최단경로는 1-2-3이 아니라 1-3이다. 즉, 호상에 다수개의 여행시간이 주어진 경우 최단경로의 부분경로는 최단경로라 보장할 수 없다.

본 논문은 이 문제의 최적해법으로 분지한계법을 제시한다. 그러나 분지한계법은 계산의 복잡도가 높아 커다란 문제를 다루는데 효율적이지 못하여 발견적기법을 개발하였다.

이 연구에서 사용되는 용어정의는 다음과 같다.

기준시간 : 호상에 있는 여행시간에 영향을 주는 단일값의 시간

시간대 : 기준시간을 크기순으로 나열할때 크기가 바로 인접된 기준시간 간의 간격, 예를들면, 기준시간이 크기순으로 T_1, T_2일때 출발시각과 T_1 (T_1은 제외), T_1과 T_2(T_2은 제외)사이의 시간간격 그리고 T_2이후의 시간을 시간대라 한다.

시간대를 고려한 Dijkstra기법 : 무환네트워크에서의 최단경로를 구하는 기법을 그대로 이용하면서 단지 마디에서의 출발시각을 고려하여 호의 여행시간만을 바꾸어주면서 계산하는 기법

2. 최적해법

이 문제를 해결하는 최적해기법으로 분지한계법을 제시한다. 이 해법은 주로 Dijkstra기법을 기본으로 한다.

분지전략으로 처음 분지점은 출발마디를 기점으로 삼아 분지를 행한다. 분지방식은 분지점(부분문제)에서 분지되어온 최종마디에 인접한 마디로 분지한다. 탐색전략은 제일 적은 하한값을 갖는 부분문제를 선택한다.

한계전략은 분지된 부분문제에 대해 가능해와 하한을 구하는 전략이다. 가능해는 출발마디부터 분지되어온 최종마디까지 시간대를 고려하여 구해진 경로값과 분지되어온 최종마디부터 목적마디까지 시간대를 고려한 Dijkstra기법으로 구해진 경로

값을 더한 값이다. 하한은 출발마디부터 분지되어 온 최종마디까지 시간대를 고려하여 구해진 경로값과 최종마디의 도착시각이 최초로 통과한 기준시간과 이후의 시간에 영향을 받는 호의 값중 각 호마다 최소값을 찾은후 분지되어온 최종마디부터 목적마디까지 Dijkstra기법으로 호별로 하나씩 찾아진 값만을 이용 최단경로의 값을 찾아 더한 값이다. 이렇게 각 부분문제에 대해 하한과 가능해를 구한후 현재의 상한과 가능해를 비교 더 작은 값을 새로운 상한으로 정한다. 두 값이 같은경우, 현재의 상한이 새로운 상한이다. 다음으로 더 좋은해를 생성할 가능성이 없는 부분문제를 제거한다. 즉 다음 조건에 있는 부분문제를 제거한다.

1) 부분문제의 하한값이 상한값보다 크거나 같은 경우

2) 출발마디부터 현 부분문제가 분지되어온 최종마디까지의 경로값이 현재의 상한을 포함하는 시간대 내에 포함되거나 이 시간대를 벗어날때 이 부분문제의 가능해가 상한보다 크거나 같은 경우

3) 분지된 경로의 최종마디가 목적마디인 경우

[정리] 출발마디부터 부분문제가 분지되어온 최종마디까지의 경로값이 현재의 상한을 포함하는 시간대에 포함되거나 이 시간대를 벗어날대 이 부분문제의 가능해가 상한보다 크거나 같은 경우 이 부분문제를 제거한다.

〈증명〉 임의의 부분문제가 제거조건 2)에 있는 경우, 이 부분문제에서의 분지가 좋은해를 생성하지 못함을 보이면 된다. 여기서 이 부분문제의 가능해는 출발마디부터 부분문제가 분지되어온 최종마디까지 시간대를 고려하여 얻어진 경로값과 분지되어온 최종마디부터 목적마디까지 시간대를 고려한 Dijkstra기법으로 구해진 경로값을 더한 값이다. 이때 가능해를 형성하는 경로상에서 분지되어온 최종마디이후의 마디는 T_i(상한을 포함하는 시간대를 T_i와 T_{i+1}사이라 가정할 때) 이후의 여행시간만을 고려하게 된다.

특히 이 부분문제의 가능해가 새로운 상한이 되기 위해서는 T_i와 T_{i+1}사이의 여행시간만을 고려해야 한다. 만약 시간대를 이용하려는 경우 이미 이 부분문제가 제거조건 2)에 있기 때문에 이 부분문제의 가능해가 현 상한보다는 더 좋은 해를 얻을 수 없다. 결국 T_i와 T_{i+1}사이의 여행시간만을 고려한 최단경로를 구해야 하는데 이 값은 이미 이 부분문제의 가능해로 얻어졌다. 그러므로 분지는 현재의 상한보다 개선된 해를 얻을 수 없다. Q.E.D

전체적으로 요약하면 처음 시간대를 고려한 Dijkstra기법으로 출발마디부터 목적마디까지의 가능해를 구하여 초기상한으로 한다. 만약 현재의 상한이 출발시각과 같은 시간대에 있으면 멈춘다. 현재의 상한이 최적이다. 그렇치 않은 경우 출발마디를 기점으로 분지를 행하면서 각 분지된 부분문제에 대해 하한과 가능해를 구한다. 동시에 각 부분문제의 가능해를 현재의 상한과 비교 수정하고 또한 제거조건에 있는 부분문제를 제거하면서 최종적으로 모든 부분문제가 제거될때 그때의 상한을 최적으로 한다.

3. 발견적 기법의 개발

3.1. 발견적 기법

발견적기법은 기존의 무환네트워크를 위한 Dijkstra기법을 수정한 것이다. 기존해법은 각 마디에 단하나의 정보 즉, 그 마디에 도달한 가장 이른 경로비용만을 고려하였다. 하지만 이 기법은 위 예제에서 보였듯이 시간대로 인하여 효율적인 해를 구할 수가 없는 경우가 있다. 그래서 수정된 기법에서는 시간대를 고려할 수 있도록 각 마디에서 정보를 다수개 유지하는 기법을 개발한다. 실제 동적계획법에 의한 완전열거식에 의해 각마디에서

모든 정보를 유지하면 이것 역시 최적해를 준다. 하지만 모든 정보를 유지하면 이에 따른 기억용량의 확보에 문제가 있게 된다. 이에 본 논문에서는 각 마디에서 모든 정보를 유지하는것이 아닌 단지 K개의 정보만을 유지하여 최적해가 아닌 효율적인 해를 얻어 이를 외판원문제의 기법인 삽입기법에 의해 효율적인 해를 개선하는 발견적기법 이다.

여기서 $d_{i,m}$($m=1, 2, 3, \cdots\cdots K$)를 마디 i에 도달한 경로비용을 시간대별로 분류 각 시간대별로 최소경로비용을 찾아 이 비용을 나열했을때 m번째로 적은 경로비용이라 하자. 이때 해법전략은 각 마디에서 $d_{i,m}$을 반복적으로 구하면서 j가 목적마디이면 멈추고 그때의 적은값이 효율적인 해이다. 여기서 고려중인 마디에 도달하는 경로비용이 K개 이상의 사간대로 분포되지 않으면 이 마디에서는 K개의 정보를 유지하지 못하게 된다. 그러나, 마디에 도달하는 경로비용이 K개 이상의 시간대로 분포되면 시간대별로 최소비용을 찾아 이중 작은 비용의 순서로 K개를 찾아 고려하는 마디에서 유지하게 된다. 만약 모든 도달비용이 같은 시간대에 있는 경우 마디는 단하나의 정보만을 유지하게 된다.

위의 방식에 의해 효율적인 경로가 얻어지면 경로의 개선을 위한 삽입을 수행한다. 현재 유지하고 있는 효율적 경로를 $s-i_1-i_2-i_3-i_4-\cdots i_p-i_{p+1-t}$라 하자. 이대 경로상에 있는 마디를 마디가 인접하도록 (s, i_1), (i_1, i_2), (i_2, i_3), $\cdots$, (i_p, i_{p+1}), (i_{p+1}, t)로 분류하여 이 분류된 두 마디 사이로 삽입가능한 마디를 한마디씩 삽입하여 경로비용을 얻은후 이 비용중 최소비용을 찾아 현재의 효율적인 경로를 구성하는 비용과 비교 개선이 되면 새로 얻어진 경로를 현재의 효율적 경로로 유지한다. 더 이상의 해의 개선이 이루어지지 않을 때까지 위의 삽입 방식을 행하게 된다.

계산절차

단계 1] 초기화

출발마디를 1이라하고 K를 각 마디에서 유지 가능한 최대 정보수라 하자.

$d_{1,1}=0$로 한다.

단계 2] 경로개선

2.1 마디 j에 도달하는 정보의 계산은

$\{d_{i,m}+d_{ij,m}\}$i는 j에 도달하는 마디

$m=1, 2, \cdots, K$

$d_{ij,m}$: $d_{i,m}$의 시간을 고려하여 선택된 호상의 여행시간이다.

2.2 유지할 정보를 선택하는데 두가지를 고려한다.

i) 도달된 비용들이 K개 이상의 시간대로 분류되어 있다.

시간대별로 최소비용을 구한후 이중 비용이 작은 순으로 K개의 정보만 유지한다.

ii) 도달된 비용들이 K개 미만의 시간대로 분포되어 있다.

시간대별로 최소비용을 구한후 그 정보만을 유지한다.

2.3를 2.1이나 2.2와 같은열에 놓는다. 마디가 j가 목적마디이면 단계 3]으로 간다. 그렇지않으면 단계 2]를 반복한다.

단계 3] 삽입단계

단계 2]까지 얻어진 경로를 개선하는 단계로 경로를 구성하는 마디사이로 한마디씩 삽입하여 개선을 시도한다. 더이상의 개선이 이루어지지 않을 때까지 행한다.

3.2. 실험설계 및 결과

발견적기법의 정확성과 효율성을 분지한계법과 비교한다. 실험내용은 마디수를 10개, 20개, 30개로 구분하고 각 마디별로 10문제씩을 만들어 각 문제별로 목적함수 값과 계산시간을 구한다. 실험은 IBM-PC XT 호환기종을 사용하였으며 언어는

QUICK BASIC으로 하였다. 실험결과는 [표 4-1]과 같다.

결과를 보면 발견적기법이 비교적 효율적인 해를 생성함을 알 수 있다. 마디수가 증가하면서 최적해와의 근접성은 줄어들지만 계산시간상에서 좋은 결과를 보여주고 있다. 우선 해의 관점에서 보았을때 발견적기법은 K=2에서 단지 문제 27을 제외하고 수렴하고 있으며 대부분의 문제에 대해 만족스러운 결과를 보이고 있다. 또 수행시간의 관점에서 보면 발견적 기법에서 K의 증가와 함께 수행시간이 어느 수준까지 소폭증가함을 보이고 있다. 그리고 발견적기법(K = 2인 경우)은 수행시간에서 최적해기법에 비해 마디 10, 20, 30 각각에서 평균 73%, 85%, 90%의 절감이 있고 평균적(모든 K를 고려)으로는 각각 평균 74%, 86%, 90%의 절감이 있음을 알 수 있다.

[표 4-1]해법간의 실험결과 비교

단위-시간 : 초

마디수	문제번호	발견적기법					최적해법
		K = 1	K = 2	K = 3	K = 4	K = 5	
10	1	14 4.89	14 6.15	14 6.64	14 6.64	14 6.64	14 11.37
	2	21 7.25	21 8.98	21 10.66	21 11.20	21 11.26	21 37.46
	3	43 5.55	32 7.25	32 8.29	32 8.29	32 8.29	32 31.42
	4	48 6.10	48 7.03	48 7.03	48 7.03	48 7.03	48 19.88
	5	53 7.75	53 9	53 9.5	53 9.5	53 9.5	53 56.57
	6	37 11.92	37 13.29	37 14.06	37 14.06	37 14.06	37 57.62
	7	14 7.47	14 9.5	14 11.43	14 12.08	14 12.03	14 26.20
	8	33 5.22	33 6.37	33 7.19	33 7.20	33 7.20	33 22.25
	9	41 9.12	41 10.55	41 11.04	41 11.00	41 11.00	41 38.06
	10	73 5.16	68 5.72	68 5.66	68 5.66	68 5.72	68 23.29
20	11	23 27.30	23 32.52	23 38.11	23 41.91	23 44.05	23 182.24
	12	24 16.97	24 19.12	24 19.50	24 19.50	24 19.50	19 58.38
	13	40 20.26	40 23.02	40 23.46	40 23.50	40 23.50	40 23.50
	14	40 24.37	40 29.5	40 33.89	40 33.83	40 33.83	40 141.76
	15	79 15.82	79 16.42	79 16.37	79 16.37	79 16.37	79 116.23
	16	42 31.03	41 35.05	41 37.68	41 37.73	41 37.68	36 506.19
	17	89 19.72	36 27.18	36 30.10	36 30.10	36 30.10	36 164.62

	18	37 28.12	37 33.94	37 33.94	37 34.65	37 34.60	37 241.02
	19	50 29.49	50 34.99	50 37.35	50 37.35	50 37.35	46 287.21
	20	51 26.36	51 21.37	51 22.25	51 22.24	51 22.25	51 135.94
30	21	43 37.52	43 42.23	43 42.26	43 45.97	43 46.03	43 611.21
	22	48 31.14	48 34.12	48 34.61	48 34.60	48 34.55	48 108.48
	23	125 35.15	110 41.58	110 41.57	110 41.63	110 41.58	65 498.95
	24	43 64.71	51 50.65	51 56.51	51 56.58	51 56.52	42 1154.48
	25	53 34.88	53 39.22	53 41.09	53 41.08	53 41.10	53 162.69
	26	43 38.33	43 44.22	43 46.69	43 46.74	43 46.74	43 109.68
	27	48 41.69	48 46.25	44 51.42	44 52.07	44 52.07	44 660.53
	28	99 37.85	99 40.15	99 40.21	99 40.20	99 40.21	99 267.93
	29	43 37.81	39 43.89	39 45.64	39 46.19	39 46.19	39 242.39
	30	53 33.12	53 34.99	53 35.04	53 34.99	53 34.99	53 301.51

* 위 표에서 상단은 경로값, 하단은 계산시간이다.

4. 결 론

본 연구는 도심지와 같은 복잡한 상황하에서 네트워크의 호를 구성하는 도로의 여건이 시간에 따라 변화하는(예를들면, 러시아워 전후) 상황하에서 두 지점간의 최단경로를 구하는 문제를 다루었다. 문제의 해법으로 최적해를 제공하는 분지한계법을 제시하였고, 무환네트워크의 Dijkstra기법과 삽입을 이용한 발견적 기법을 개발 하였다.

최종적으로 세 기법을 계산시간과 목적함수값에서 검토하였다. 그결과 해는 비교적 만족스러운 결과를 보였으며 계산시간상에서 발견적기법이 최적해기법에 비해 마디 10, 20, 30 각각에서 평균 74%, 86%, 90%의 절감이 있음을 보였다.

추후 연구과제로는 환이있는 네트워크에 대한 문제의 특성과 해법의 개발에 대한 연구가 필요하다.

참고문헌

[1] 박순달, OR(경영과학), 대영사, 1987

[2] Aneja, Y. P., V. Aggarwal and K.P.K. Nair, "Shortest Chain Subject to Side Constraints", Networks, Vol. 13, pp. 295-302, 1983

[3] Beasley and N. Chrisofides, "An Algorithm for the Resource Constrained Shortest Path Problem", Networks, Vol. 19, pp. 379-394, 1989

[4] Bodin, L., B. Golden, A. Assad and M. Ball, "Routing and Scheduling of Vehicles and Crews ; the state of the Art", Computers & Opes. Res, Vol. 10, pp. 180-211, 1983

[5] Halpern, J, and I. Priess, 'Shortest Path with Time Constraints on Movement and Parking", Networks, Vol. 4, pp. 241-253, 1974

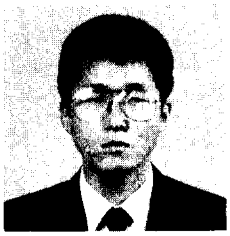
[6] Hu, T. C., Integer programming and Networks Flows, Addison-Wesley Publishing Company, 1970

[7] Ibaraki, T., "Algorithms for Obtaining Shortest Path Visiting Specified Nodes", SIAM Rew, Vol. 15, pp. 309-317, 1973

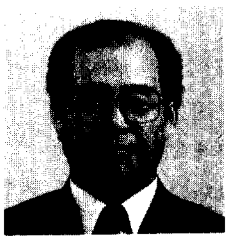
[8] Katoh, N., T. Ibaraki, and H. Mine, 'An Efficient Algorithm for K Shortest Simple Paths", Networks, Vol. 12, pp. 411-427, 1982

[9] Phillips, D. T., Fundamentals of Network Analysis, Prentice-Hall, Inc, 1981

저자소개



저자(이명석)은 87년 경희대 산업공학과를 졸업 하였고 90년 서울대학교 대학원 산업공학과를 졸업하였다. 경영과학 및 컴퓨터 응용분야에 관심을 가지고 있다.



저자(박순달)은 현재 한국경영과학회 회장이며, 서울대학교 산업공학과 교수로 재직중이다. Univ of Cincinnati에서 이학박사 학위를 획득하였고, 독일 Rubr University에서 연구원으로 근무하기도 했던 저자는 Deterministic O.R. 분야와 컴퓨터 활용에 흥미를 가지고 연구를 하고 있다.