Journal of the Korean Institute of Industrial Engineers
Vol. 16, No. 2, Dec., 1990

공공차량 경로문제의 이중 최단나무 결합 해법

장 병 만*

Double Shortest Arborescence & Merging Algorithm for the Public Vehicle Routing Problem

Byung-Man Chang*

Abstract

In this paper, the Double Shortest Arborescence & Merging method is presented as an efficient heuristic algorithm for the Public Vehicle Routing Problem which is to find the minimum total cost routes of M or less vehicles to traverse the required arcs(demand streets) at least once and return to their starting depot on a directed network.

Double Shortest Arborescence which consists of forward shortest aborescence and backward one informs M or less shortest routes to traverse all required arcs. The number of these routes is reduced to M or less by merging routes.

The computational experiment based on randomly generated netwokrs reports that this algorithm is efficient.

1. 서 론

본 연구에서는 공공차량 운행문제(Public Vehicle Routing Prblem : PVRP) [3]의 효율적인 발견적 해법을 제시하고자 한다.

PVRP는 유방향의 연결 네트워크 G(N, A)에서 M대 이하의 차량으로 출발지점 s를 떠나 서비스가 요구되는 수요호의 집합 R(⊆ A)를 적어도 1번씩은 방문하고 다시 출발지점으로 돌아오는데 소요되는 차량들의 고정비와 운행비용의 합이 최소인 각 차량별 운행경로를 찾는 문제이고 그 특성은 표 1과 같다.

* 서울산업대학 산업공학과

적용분야로는 도로 청소차, 쓰레기 수거차, 경찰 순찰차, 우편물 수집차, 가로차(street sweeper), 학교 통학버스, 선로 점검수리원 등의 경로문제나 복수 우체부 경로문제 등이 있다.

본 논문에서 사용하는 용어는 다음과 같다.

A : 유방향 호(Arc)들의 집합 $m=|A|$

R : 수요호(Required Arc)들의 집합 $r=|R|$

$\bar{R}$: 비수요호(Nonrequired Arc)들의 집합 $\bar{R}=A-R$

N : 마디(Node)들의 집합, $n=|N|$

C_{ij} : 마디 i에서 마디 j로 가는 호의 비용

s : 출발점

t : 종착점

M : 최대 투입 차량수

$G(N, A)$: N과 A로 이루어진 네트워크

F : 차량의 고정비

x_{ij} : 호(i, j)사이의 흐름량, 통과 차량수

LP_{ij} : 호(i, j)를 지나는 s와 t간의 최단거리.

SP_{ij} : 호(i, j)를 지나는 s와 t간의 최단경로.

PVRT의 최적 분지한계 해법[3]으로 현실적인 문제를 풀게되면 계산시간과 노력이 엄청나게 증가함으로 이를 줄이기 위해 좋은 상한이나 초기해 값을 구할 필요가 있고, 또한 현실적인 연건속에서는 좋은 최적 근사해도 적용할 수가 있기 때문에 PVRP의 발견적 해법을 연구해 보고자 한다.

이중 최단 나무결합 해법(Double Shortest Arborescence & Merging Algorithm : DSAM법)이라는 DSAM법에서는 먼저 출발점 s에서 모든 각 마디까지의 최단거리와 각 마디에서 종착점 t까지의 최단거리를 구한 후, 각 수요호를 지나는 s에서 t까지의 최단 경로를 모든 수요호에 대해서 구한다. 그 다음에 이 최단 경로들을 운행비용의 내림차순으로 선택하되, 수요호 집합 R이 모두 통과되는 최소한의 경로를 선택한다. 여기서 경로수가 M개 이하이면 가능해를 구한 것이고, 그렇지 못하면

표 1. PVRP의 특성

1. 모기지 수	1개소	
2. 차량수	복수 차량	M대
3. 차 종	동일 종류	
4. 수요 형태	확정적	
5. 수요 위치	일부의 호(Arc), 도로	$(i,j)\in R\subseteq A$
6. 네트워크 형태	유방향	
7. 경로의 연속 운행시간	제한 없음	
8. 도착 및 서비스시간	지정 없음	
9. 작업내용	배달, 상차, 하차, 상하차	
10. 비 용	운행비용, 운행거리	
11. 목적함수	총 운행거리의 최소화	(C_{ij})
	(필요 차량수 최소화 도모)	
12. 해	M^* 대의 차량의 최적 운행경로	$M^*<M$

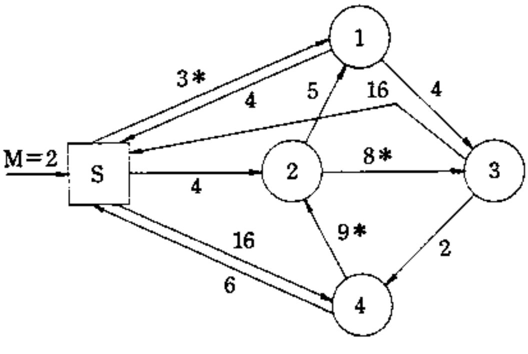
유일하게 통과하는 고유의 수요호의 수가 적은 경로를 해체하고, 이 고유의 수요호들을 인접 경로에서 우회하여 통과하도록 다른 경로속에 결합(merge) 시키는 과정을 반복하여 M개 이하의 최소 총비용의 경로를 찾는다.

PVRP는 호경로문제(Arc Routing Problem)의 한 종류이며, 지역 우체부문제(Rural Postman Problem)[5, 9]의 일반화된 모형이고, 복수 우체부문제(Multiple Chinese Postman Problem)[2]과 유용량 우체부 문제(Capacitated Chinese Postman Problem)[9], 유용량 호경로문제(Capacitate Arc Routing Problem : CARP)[4]와 관련이 깊은 모형이다.

PVRP의 연구는 장병만, 박순달[3]이 PVRP의 수리모형을 제시하고, 최소비용 해법으로 부분문제를 푸는 최적의 분지한계 해법을 개발한 것이있다.

2. 이중 최단나무 결합 해법

PVRP의 네트워크 G(N, A)는 그림 1과 같다.



경로 1 : S→1→3→4→2→1→S
경로 2 : S→2→3→4→S
* : 수요호
F(차량고정비)=10

그림 1. PVRP의 네트워크 예

PVRP의 가정은 다음과 같다.

1. 투입하는 차량의 수는 M이하이며, 차량의 종류는 동일하다.

2. 모든 호들은 1회 이상 통과할 수 있으며, 서비스할 때나 서비스 안할 때의 통과시간은 같다.

2-1. 이중 최단거리 나무

각 수요호를 지나는 s에서 t까지의 최단 경로를 효과적으로 구하기 위해서 이중 최단거리 나무(Double Shortest Arborescence : DSA)를 이용하고자 한다. 이 DSA를 구하기 위하여 다음과 같은 정의를 한다.

정의 1. 최단거리 나무(Shortest Distance Arborescence : SAD)

출발마디(점) s에서 모든 다른 마디들까지의 최단 경로들을 유방향 네트워크 상에서 구했을 때 나오는 이 최단 경로들로 구성되고 마디 s를 뿌리로 하는 유방향 극대나무(Spanning Arborescence).

최단거리 나무는 Dijkstra의 해법으로 $O(n^2)$의 계산시간 복잡도로 아주 효율적으로 풀 수 있다.

그림 2에서의 최단거리 나무는{a, b, d, f, i} 이다.

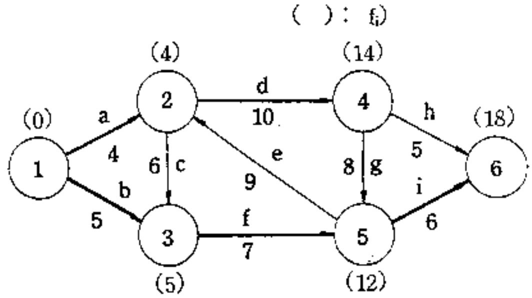


그림 2. 이중 최단거리 나무

정의 2. 이중 최단거리 나무(Double Shortest Arborescence)

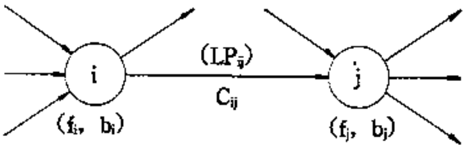
유방향 연결 네트워크 상에서 출발점 s에서 종착점 t까지 전방향 최단거리 나무를 Dijkstra법으로 구하고, 같은 방법으로 종착점 t에서 출발점 s까지 역방향으로 최단거리 나무를 구하여 결합시킨 나무.

이 이중 최단거리 나무에서는 s에서 각 마디 j까지의 최단거리 f_j와, 각 마디 j에서 t까지의 최단거리 b_j가 구해지므로, s에서 마디 j를 거쳐 t로 가는 최단경로 SP_j가 구해지고, 이 때의 최단경로의 길이 LP_j는 f_j+b_j이다.

또 이 이중 최단거리 나무로 출발점 s에서 호(i, j)를 지나 종착점 t로 가는 최단경로와 그 길이를 구할 수 있고, 이때 길이는 $LP_{ij}=f_i+c_{ij}+b_j$이다.

정리 1. 출발점 s에서 호(i, j)를 지나 종착점 t까지 가는 경로의 길이가 LP_{ij}이면 이 경로는 호(i, j)를 지나는 경로들 가운데 최단경로이다. 단, $LP_{ij}=f_i+c_{ij}+b_j$이다.

증명) 이중 최단거리 나무에서 호(i, j)의 시작마디 i와 도착마디 j에 대해서 Dijkstra법으로 구한 Permanent Label이 각각(f_i, b_i), (f_j, b_j)라면, 출발점 s와 마디 i간의 최단경로는 f_i, 마디 j와 종착점 t간의 최단경로는 b_j이고, 호(i, j)의 최단 길이가 C_{ij}이다.



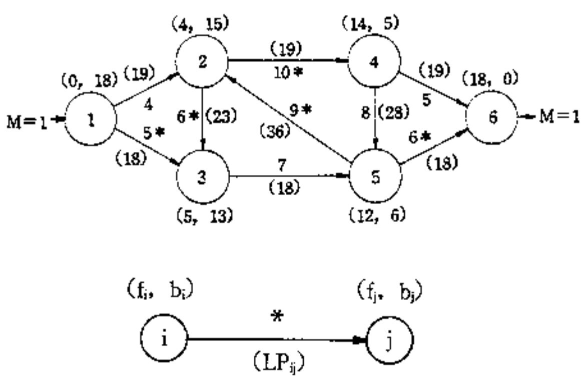
그러므로 호(i, j)를 지나는 s와 t사이의 최단 경로의 길이는 $f_i+b_j+C_{ij}$ 즉, LP_{ij}이다.

그러므로 PVRP에서 거쳐가야 하는 수요호(i, j) ∈R를 지나는 최단 경로들과 그 길이들을 모두 이중 최단거리 나무로 구할 수 있다.

이들 경로들을 길이의 내림차순으로 나열하면, 수요호의 수 R개 이내의 다수의 경로들을 구할수 있게 된다.

이중 최단거리 나무의 방법으로 그림 2에서 각 호를 지나는 최단 경로를 찾아보자.

먼저 그림 3과 같이 이중 최단거리 나무를 작성한다.



*: 수요호(i, j) 표시

(LP_{ij}): 호(i, j)를 지나는 최단경로 길이

(f_i, b_i): 마디 i의 Permanent Label

그림 3. PVRP의 이중 최단거리 나무

그 다음에 각 호(i, j)를 거쳐가는 최단경로 SP_{ij}와 길이 LP_{ij}계산한다. 예를 들어 호(5, 2)를 지나는 경로의 길이의 계산은 다음과 같다.

$LP_{52}=12+9+15=36$

$SP_{52}=1-3-5-2-4-6$

그리고 각 수요호를 지나는 최단경로를 작성한다.

경로 1, SP_{52}: $1\overset{*}{-}3-5\overset{*}{-}2-4-6$ (36)

경로 2, SP_{45}: $1-2-4-5\overset{*}{-}6$ (28)

경로 3, SP_{23}: $1-2\overset{*}{-}3-5\overset{*}{-}6$ (23)

경로 4, $SP_{13}=SP_{56}$: $1\overset{*}{-}3-5\overset{*}{-}6$ (18)

( ) : LP_{ij}길이

2-2. 경로의 축소와 결합

PVRP에서는 M개의 최단경로들 속에 R개의 수요호들이 모두 포함되어야 한다. 그런데 이중 최단거리 나무법으로 구한 이 수요호들을 지나는 최단경로들은 R개 이내이지만 보통 M개 이상이다. 또한 짧은 경로들 가운데는 긴 경로들 속에서 통과되는 수요호들만 가지고 있어서 이 짧은 경로가 불필요한 경우가 있을 수 있다. 이러므로 경로들간의 결합(Merging)을 시켜서 M개 이하로 축소시켜야 한다.

최단경로 수의 축소는 다음과 같은 방법으로 실시한다.

우선 수요호가 하나라도 포함되어 있는 경로들의 운행거리의 길이의 내림차순으로 일련 번호를 부여한다. 가장 긴 경로의 번호가 1이 된다.

번호순으로 점검하면서 지금까지의 긴 경로속에서 현재의 경로속의 모든 수요호들이 모두 서비스되는지 검토하여 모두 앞에 나온 긴 경로속에서 서비스되면 현재의 경로를 제거한다. 즉 경로 1에서 경로 i까지의 경로속에 경로 i+1번째의 수요호들이 모두 서비스되면 경로 i+1은 제거해도 된다.

가장 긴 경로를 기준으로 짧은 경로를 제거시키는 이유는 다음과 같다. 가장 긴 경로는 어떤 수요호들을 지나는 최단 경로이기 때문에, 이 경로는 제거할 수 없으므로 기준으로 두는 것이다. 또한 긴 경로는 짧은 경로에 비하여 대부분 많은 수요호들을

서비스하며 통과하기 때문이다.

이상과 같은 방법으로, 경로 1을 기준으로 경로 1에 포함된 모든 수요호들을 다른 모든 짧은 경로 속에서 지우고 이 때 어떤 경로의 모든 수요호들이 다 지워진 경로는 제거한다. 다음에는 경로 2를 기준으로 경로 3 이하에서 경로 2의 수요호들이 포함되어 있으면 지우고, 모든 수요호들이 다 지워진 경로들은 제거한다. 이런 절차로 경로 i을 기준으로 경로 i+1 이하에서 경로 i의 수요호들이 포함되어 있는 것을 지우므로 경로 1에서 경로 i 까지의 수요호들을 경로 i+1부터 그 이하의 경로 상에서 다 지우고, 모든 수요호들이 다 지워진 경로들은 제거하는 방법을, 고유의 수요호를 가지는 하나의 경로만 남을 때까지 반복한다. 그런데 이중 최단거리 나무에서 구한 R개의 수요호를 지나는 최단경로들을 만들면 비수요호는 수요호간의 최단 경로상에만 들어가므로, 비수요호 때문에 불필요하게 위회하는 현상이 배제되며, 모든 경로들은 고유의 수요호를 통과하므로 하나의 짧은 경로속에서 보다 긴 다른 경로속의 수요호들이 포함되어 서비스되는 경우는 일어나지 않는다.

이렇게 해서 나온 경로의 수가 M 이하이면 이것은 하나의 가능해로 삼을 수 있다. 그러나 일반적으로 M 이상인 경우에는 이 경로들을 결합시켜 경로수를 축소시키는 방법을 찾아보아야 한다.

현재 구해진 경로의 수가 NP(>M)개라면, 이것을 M개 이하로 만들기 위해 경로들간의 결합을 시켜야 한다.

경로의 결합에서 고려할 사항은 다음과 같다.

첫째로, 경로의 수는 결합시켜서 M개 이하로 줄어져야 한다.

경로의 수가 M개까지는 운행비용이 크게 증가되므로 무조건 경로간의 결합을 시켜야 한다.

둘째는, 차량이나 우체부 투입에 따른 고정비가 경로수에 비례하여 증가하므로 경로수를 줄여야 한다. M대 이하의 경로간의 축소에는 각 경로별로 차량의 고정비 F를 고려하여 총 운행비용과 총 고정비의 합인 총비용이 최저가 되는 경로의 수, 곧 차량수 M^*(<M)와 경로들을 찾아야 한다.

이 때는 경로의 결합으로 운행비용은 증가하나 고정비가 줄어들게 되므로 총비용이 최저가 되는 균형점(trade off point)을 구해야 한다.

셋째는, 가장 적은 수의 고유의 수요호들을 가지는 짧은 경로를 해체하고 그 수요호별로 삽입할 경로를 찾도록 한다. 그 이유는 경로와 다른 경로를 결합하면 유방향 네트워크상에는 우회하는 일이 많아서 나쁜 해가 얻어지기 때문이다.

그러므로 경로의 결합방법은 다음과 같게 한다.

우선 해체할 경로를 하나 찾는다. 이 경로로는 고유의 수요호의 수가 적은 경로이면서 길이가 긴 경로를 우선적으로 한다. 해체할 경로의 수는 NP−M개이다. 그리고 해체된 경로 BR의 수요호 각각에 대해서 이 수요호를 삽입시킬 경로 NR과 그 위치를 찾아야 한다. 이 때 먼저 모든 마디간의 최단거리를 구해야 한다. 이를 위해 Moffat 해법[8]을 이용하면 $O(n^2 \log n)$의 계산시간 복잡도가 소요된다.

그 다음 삽입시킬 수요호(i, j)에 대해 해체안된 경로상의 삽입위치 마다 p, q를 찾아본다. 여기서 하나의 기존 경로에서 호(i, j)가 삽입될 때의 우회함으로 증가되는 비용 π_{ij}(Penalty Cost)는 다음과 같다.

$$\pi_{ij}=\text{Min}(C_{pi}+C_{ij}+C_{jq}-C_{pq})\ \forall p,\ q\in NR$$
$$(p,\ q)\in NR \quad (i,\ j)\in BR$$

여기서 p, q는 같은 마디(p=q)이거나, 비수요호 (p, q)이거나, 비수요호가 이어진 부분경로 p−q일 수 있다. 수요호(i, j)∈BR을 삽입하려는 위치에 따라서 3가지의 경우가 나올 수 있다.

첫째, 기존 경로에서 마디 P 다음에는 수요호가 없다면 호(i, j)를 지나는 p와 t사이의 최단경로 p−i−j−t를 마디 p 다음에 이어주면 된다.

둘째로, 삽입될 수요호(i, j)가 기존 경로의 출발점과 처음 수요호(q, v)사이에 들어간다면, 호(i, j)를 지나는 s와 q사이의 최단경로 s−i−j−q를 기존 경로의 마디 q앞에 연결시킨다.

셋째로 호(i, j)∈BR이 기존 경로의 2개의 수요호 (o, p)와 (q, v)사이에 삽입된다면 호(i, j)를 지나는 p와 q사이의 최단 경로 p−i−j−q를 기존 경로의 마디 p와 q사이에 연결시킨다.

그 다음 NP<M 이하가 되면 고정비 F를 고려한 목적함수 값 Z'를 계산하여 경로수가 하나 많을때의 총비용 Z와 비교하여 $Z \leq Z'$일 때까지 경로결합을 계속한다.

2-3. 이중 최단나무 결합 해법

이중 최단나무로 구한 R개 이내의 최단경로를 축소, 결합시켜 양호한 가능해를 구하는 해법을 이중 최단나무 결합해법(Double Shortest Arborescence & Merging Algorithm: DSAM법)이라 하자.

DSAM법의 절차는 다음과 같다.

단계 1. 초기화 K=0

단계 2. 이중 최단거리 나무(DSA)를 구한다.

(1) Dijkstra법으로 전방향의 최단거리 나무를 구하여 각 마디 i마다 출발점 s에서의 최단거리 f_i를 계산한다.

(2) 역방향의 최단거리 나무를 구하여 각 마디 i에서 종착점 t까지의 최단거리 b_i를 계산한다.

(3) 각 호(i, j)를 지나는 최단경로(SP_{ij})의 길이 (LP_{ij})를 계산한다. … $LP_{ij}=f_i+C_{ij}+b_j$

단계 3. 길이의 내림차순으로 최단경로들을 나열한다.

단계 4. 경로수의 축소

(1) k=k+1, 경로 k의 수요호들은 경로 k+1 이후의 모든 경로에서 지운다.

(2) 모든 수요호들이 지워진 경로는 제거한다.

(3) 확정된 경로수(NP)를 조정한다.

(4) 검토안된 경로수 L<1이면 운행비용의 합 Z계산, 단계 5로 간다.

L>2이면 단계 4−(1)로 간다.

단계 5. 경로의 해체와 결합

(1) NP>M이면 고유의 수요호가 가장 적은 경로 1개를 해체하기 위해 선택한다.

NP<M이면 단계 6으로 간다.

(2) 선택된 경로의 수요호(i, j)∈BR들에 대한 최소 우회비용 π_{ij}과 삽입위치(p, q)를 계산한다.

$$\pi_{ij}=\mathrm{Min}(C_{pi}+C_{ij}+C_{jq}-C_{pq}) \ \forall \ p, \ q \in NR$$

단, p, q는 p=q, 비수요호(p, q) 또는 비수요호들의 부분경로 p−q 이다.

(3) 수요호(i, j)∈BR들을 삽입위치에 넣어 다른 경로들과 연결한다.

(4) 전체 운행경로 비용 Z'를 계산한다.

(5) 형성된 경로의 수를 파악한다.

NP<M이면, 단계 6으로 간다.

$NP \geq M$이면 Z=Z', 단계 5−(1)로 간다.

단계 6. 총비용과 최종해 도달을 판정한다.

(1) Z'=Z'+(NP×F), Z<Z'이거나 NP=1이면 단계 6−(3)으로 간다.

(2) Z=Z', 단계 5−(1)로 간다.

(3) Z=Z', 최종해를 찾았다. STOP.

각 경로별 코스와 총비용 Z를 출력한다.

정리 2. r=|R|이고 r<n이라 가정하면, 이중 최단거리 결합해법의 계산시간 복잡도는 $O(n^3)$이다.

증명) 단계 2와 3에서 DAS 복합나무를 구하는 데는 $O(n^2)$이고, 단계 4에서 한경로의 수요호들을 찾는데 $O(r^2)$이고, 각 경로상의 모든 수요호를 찾는데는 이것은 r배만큼 복잡도가 늘어나므로 $O(n^3)$ 이내이며, 단계 5에서 $O(n^2)$만에 수요호가 적은 경로를 찾으며, 단계 5에서는 모든 마디간의 최단거리가 계산되어야 하므로 $O(n^2 \log n)$이다. 그러므로 전체 계산시간 복잡도는 $O(n^2)+O(n^3)+O(n^2)+O(n^2 \log n)=O(n^3)$이다.

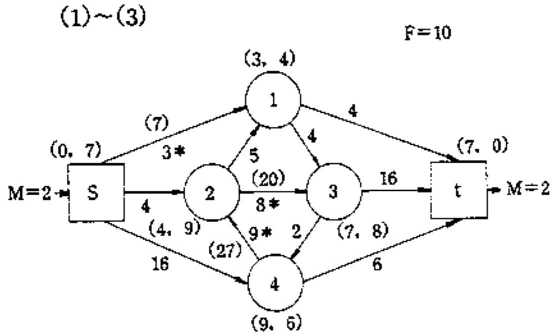
3. 적용 예제

그림 1의 PVRP 네트워크 예에 대하여 이중 최단나무 결합해법(DSAM법)을 적용하여 풀어보자.

단계 1. 최기화 $k=0$

단계 2. 이중 최단거리 나무를 작성.

(1)~(3)



단계 3. 내림차순으로 경로들을 나열

경로번호	경　　로	길이
경로1	$s \overset{*}{\rightarrow} 1 \rightarrow 3 \rightarrow 4 \overset{*}{\rightarrow} 2 \rightarrow 1 \rightarrow t$	27
경로2	$s \rightarrow 2 \overset{*}{\rightarrow} 3 \rightarrow 4 \rightarrow t$	20
경로3	$s \overset{*}{\rightarrow} 1 \rightarrow t$	7

단계 4. 경로수의 축소

(1)~(2)

$K=1$ 경로 1. $s \overset{*}{\rightarrow} 1 \rightarrow 3 \rightarrow 4 \overset{*}{\rightarrow} 2 \rightarrow 1 \rightarrow t$

경로 2. $s \rightarrow 2 \overset{*}{\rightarrow} 3 \rightarrow 4 \rightarrow t$

경로 3. $s \overset{*}{\rightarrow} 1 \overset{*}{\rightarrow} t$는 경로 1에서 통과됨으로 제거

(3) $Z=27+20=47$

$NP=2$. $L=1$. 단계 5로 간다.

단계 5. 경로의 해체와 결합.

(1) $NP=2=M$

경로 2가 고유수요호(2, 3) 한개 뿐이므로 해체함

(2) π_{23}의 계산

호(2, 3)이 경로 1에 삽입될 위치는 마디 1과 4 사이 또는 마디 2와 t사이이다.

$$\begin{aligned}\pi_{23} &= \min(C_{12}+C_{23}+C_{34}-C_{14},\ C_{22}+C_{23}+C_{3t}-C_{2t}) \\ &= \min(15+8+2-6,\ 0+8+8-9) \\ &= \min(19,\ 7) \\ &= 7\end{aligned}$$

(3) 그러므로 호(2, 3)을 지나는 마디 2와 t사이의 최단 경로가 마디 2에 연결되어야 한다.

(4) $Z'=27+7=34$

(5) $NP=1<M=2$, 단계 6으로 간다.

단계 6. (1) $Z'=34+10\times1=44$, $Z=47+10\times2=67$

$Z'<Z \rightarrow Z=Z'=44$

(3) 최종해

$M=1$

경로 : $s \overset{*}{-} 1-3-4 \overset{*}{-} 2 \overset{*}{-} 3-4-t$

$Z=44$

4. 계산결과의 분석

PVRP의 최소비용 문제를 이용한 최적해법과 이중 최단나무 결합해법에 대한 계산상의 효율성의 비교 분석을 하고자 한다.

PVRP의 네트워크를 생성시키기 위하여 네트워크 생성에 있어서 Benchmark프로그램인 Klingman[6]의 NETGEN을 사용하였다. 이 NETGEN을 이용하여 호의 수가 10개부터 1000개까지의 네트워크를 200개 생성을 시켰는데, 이 가운데서 불법경로가 발생하는 네트워크는 20개가 생겼으며, 이 네트워크의 크기는 표 2와 같다.

호의 밀도는 10%, 25%, 40%의 3가지로 하였다. 개선해법의 프로그램 코드는 Microsoft사에서 개발한 QuickBasic 4.0으로 프로그램되고, 실험은

표 2. 네트워크의 구조

문제	문 제 크 기			
	호	마디	수요호	투입차량수
1	13	7	2	3
2	16	8	2	3
3	16	8	2	3
4	16	8	3	3
5	16	8	3	3
6	18	9	4	3
7	20	10	3	3
8	20	10	5	3
9	20	13	6	3
10	22	10	6	3
11	22	11	5	3
12	30	17	7	5
13	50	20	12	5
14	50	22	15	5
15	70	20	12	10
16	80	22	20	10
17	90	31	17	3
18	94	31	20	3
19	100	24	23	10
20	120	30	29	10

IBM-PC AT-286 시스템에서 실행하였다.

PVRP의 최소비용 해법은 장병만, 박순달[3]을 이용하였다. 이 최적해법은 PVRP의 최소비용 문제 모형에서 불법경로 조건을 완화시킨 모형이 Totally Unimodular이므로 이 모형을 부분문제로 하여 최소비용 해법으로 풀어서 해를 구하고, 이 해 가운데 불법경로가 나오면, 불법경로에 인접하여 진입할 수 있는데 $X_{ij}=0$인 비수요호(i, j)의 흐름 하한을 $l_{ij}=1$로 조정하여 부분문제를 생성시키고, 이를 다시 최소비용 해법으로 풀어서 최적해를 구해나가는 분지한계 해법이다.

이중 최단나무 결합해법의 최적해법에 대한 해의 정확도와 계산시간의 효율성을 비교분석한 결과가 표 3과 같다.

표 3. 이중 최단나무 결합 해법과 최적 해법의 효율성 비교

시간 : 분

문제	최적 해법		최단나무 결합해법		
	Z^*	시간	Z_{DSA}	Z_{DSA}/Z^*	시간
	34	0 : 05	34	1.000	0 : 01
2	20	0 : 05	20	1.000	0 : 01
3	33	0 : 12	33	1.000	0 : 01
4	24	0 : 02	24	1.000	0 : 01
5	28	0 : 02	28	1.000	0 : 01
6	55	0 : 14	55	1.000	0 : 01
7	466	0 : 17	466	1.000	0 : 02
8	289	0 : 44	289	1.000	0 : 01
9	843	0 : 17	843	1.000	0 : 03
10	269	0 : 25	269	1.000	0 : 03
11	458	0 : 32	459	1.000	0 : 02
12	982	0 : 43	1124	1.145	0 : 05
13	1552	0 : 15	1567	1.009	0 : 09
14	3398	12 : 04	3398	1.000	0 : 18
15	867	5 : 33	923	1.065	1 : 14
16	8155	1 : 06	9847	1.207	0 : 56
17	1902	3 : 30	1951	1.026	0 : 39
18	2188	9 : 15	2311	1.056	0 : 53
19	9155	2 : 43	9487	1.036	1 : 22
20	5679	10 : 02	6284	1.106	2 : 15

해의 정확도는 표 3의 자료로 평균치 분석을 하면, Z_{DSA}를 이중 최단나무 결합해법의 해의 값이고, Z^*는 최적값이라 할 때 해의 평균 오차가

$$(Z_{DSA}-Z^*)/Z^*=0.03$$

이다. 평균적으로 볼 때 해의 정확도가 좋다고 추정된다.

계산시간은 평균치 분석결과, 최적해법의 15%

이내의 시간이 소요되는 것으로 나타났다. 이중 최단나무 결합해법은 네트워크의 구조에 따라서 속도가 크게 빠르지는 못한 경우도 있지만 상당히 양호한 발견적 해법으로 사료된다.

5. 결 론

PVRP의 효율적인 발견적 해법으로 $O(n^3)$의 이중 최단나무 결합해법을 제시하였다. 특히 차량의 투입 고정비와 운행비용의 합인 총비용을 최소화 시키는 해법으로써, 수요호를 지나는 M개 이하의 최단경로를 찾는 문제로 PVRP를 변환시켰고, 최단거리 나무를 이중으로 결합시켜 각 수요호를 지나는 최단경로들을 구하여 M개 이하가 되도록 경로간의 축소와 결합을 시키는 해법이다.

추후 연구방향으로는 이 발견적 해법의 효율성을 제고하거나, 특히 경로(Path, Route)나 나무(Tree, Arborescence)의 특성을 살려서 푸는 해법이나, 분지한계법을 뛰어넘은 효율적인 최적해법 연구를 하는 것이다. 또한 차량의 운행조건이 첨가된 실제적인 유용량 호경로문제(CARP)의 좋은 해법을 개발하는 것도 큰 의미가 있겠다.

참고문헌

[1] 박순달, OR(경영과학), 대영사, 1987.

[2] 장병만, "복수지역 우체부 문제", 서울대학교 박사논문, 1989.

[3] 장병만, 박순달, "공공차량 경로문제 해법 연구", 한국경영과학회지, 제 14권 2호, 53-66, 1989.

[4] Bodin, L., Golden, B., Assad, A. and Ball, M., "Routing and Scheduling of Vehicles and Crews", Comput. & O.R. 10, 63-211, 1983.

[5] Frederickson, G., "Approximation Algorithms for Some Postman Problems", JACM, 26, 538-554, 1979.

[6] Klingman, D., Napier, and Stutz, J., "NETGEN : A Program for Generating Large Scale Capcaitated Assignment, Transportation and Minimum Cost Flow Network Problems", Management Science, 20, 5, 814-821, 1974.

[7] Levy, L., "The Walking Line of Travel Problem : An Application of Arc Routing and Partitioning", Ph. D. Dissertation, University of Maryland, 1987.

[8] Moffat, A. and Takaoka, T., "An All Pairs Shortest Path Algorithm with Expected Time $O(n^2 \log n)$", SLAM '. Comput. 16, 6, 1023-1031, 1987.

[9] Orloff, C., "A Fundamental Problem in Vehicle Routing", Networks, 4, 35-64, 1974.

[10] Pearn, W.L., "The Capacitated Chinese Postman Problem", Ph. D. Dissertation, University of Maryland, 1984.