

시간단위 차량통행 속도정보의 활용을 위한 구역분할 방법의 연구

박성미 · 문기주[†]

동아대학교 산업경영공학과

A Study on Area Division Method to use the Hour-based Vehicle Speed Information

Sung Mee Park · Geeju Moon[†]

Dept. of Industrial and Management Systems Engineering, Dong-A University

This research is about developing an efficient solution procedure for the vehicle routing problem under varying vehicle moving speeds for hour-based time interval. Different moving speeds for every hour is too difficult condition to solve for this type of combinatorial optimization problem. A methodology to divide the 12 hour based time interval offered by government into 5 different time intervals and then divide delivery area into 12 small divisions first and then re-organizing them into 5 groups. Then vehicle moving speeds are no longer varying in each of the 5 divisions. Therefore, a typical TSP solution procedure may be applied to find the shortest path for all 5 divisions and then connect the local shortest paths to form a delivery path for whole area. Developed solution procedures are explained in detail with 60 points example.

Keywords : Vrp(Vehicle Routing Problem), Tsp(Traveling Salesman Problem), Combinatorial Optimization, Shortest Path

1. 서 론

공급사슬에서 물류의 비중이 점진적으로 커지고 그 비용 또한 증가하기 때문에 각 기업들의 물류비 절감 노력이 계속되고 있다. 기업의 물류비 절감의 필요성과 늘어난 물동량에 비해 부족한 도로사정으로 인한 정체는 보다 효율적인 배송을 위한 연구를 필요로 하게 만들었다. 배송문제 즉, 차량경로문제(VRP : Vehicle Routing Problem)는 NP-hard 문제로 수요지의 수가 증가하고 현실적인 다양한 제약들이 포함되게 되면 최적해를 구하기가 매우 어렵기 때문에 다양한 배송 환경 하에서의 유효한 해를 구하기 위해 많은 연구들이 수행되고 있다.

본 연구의 대상인 시간구간별로 배송차량의 이동속도가 변하는 문제는 일반적인 TSP와는 달리 방문지점들 사이의 거리가 이동속도가 바뀌므로 하여 특정지점을 이동하게 되는 시점에 따라 두 지점사이의 거리가 가변적인 것과 같은 상황이 되어 효율적으로 해를 탐색하는 해법의 개발이 용이하지 않다.

이동하는 시간대에 따라 차량의 이동속도가 변동하는 경우를 효율적으로 해결하기 위해 본 연구에서는 B시에서 조사하여 제공하는 차량이동속도 정보를 분석하여 배송대상구역을 차량의 속도 변화가 없는 5개의 구역으로 분할하여 분할된 구역별로 배송해야할 순서를 결정하는 방안을 개발하였다.

논문접수일 : 2010년 11월 19일 게재확정일 : 2010년 12월 03일

[†] 교신저자 gjmoon@dau.ac.kr

* 이 논문은 동아대학교 교내연구비 지원에 의하여 연구되었음.

5개로 분리되어 해당 구역내의 배송지들을 이동시에는 차량의 속도변화가 없으므로 일반적인 TSP로 구하고자 하는 최적해를 탐색할 수 있도록 하였는데, 출발점으로 돌아오지 않는 두지점사이의 최단경로를 탐색한 후 5개 구역의 경로들을 연결함으로써 전체의 최단경로 탐색이 가능하게 하였다.

2. 가변이동속도 차량경로문제의 특성

2.1 문헌연구

차량경로문제는 TSP에 무게제약, 시간제약, 이기종 차량 등 여러 제약이 혼합된 문제로 볼 수 있는데 실제로 차량경로문제는 많은 제약이 따르고 그 제약에 따라 CVRP (Capacitated Vehicle Routing Problem), HVRP(Heterogeneous fleet Vehicle Routing Problem), MDVRP(Multiple Depot Vehicle Routing Problem), SDVRP(Split Delivery Vehicle Routing Problem), VRPTW(Vehicle Routing Problem with Time Window), VRP VRPDT(Vehicle Routing Problem with Due Time) 등으로 나누어 연구되고 있다.

차량경로문제는 TSP로 수학적 모형화는 가능하나 최적해를 구하는 다항식 알고리즘이 존재하지 않는 NP-hard 문제로 분지한계법 혹은 기타 발견적기법으로 문제를 해결하고 있다[9]. Dantzig and Ramser[3]가 LP(Linear Programming)를 이용한 발견적 해법을 개발하였고, Clarke and Wright[4]는 각 목적지 쌍들의 조합을 변화시켜 더 나은 해를 찾는 절약(saving)의 개념을 이용한 효과적인 발견적 기법을 개발하였다. Christofides and Elion[11]은 차량의 최대 운행거리 제한에 대한 제약조건을 추가시킨 문제를 다루면서 라그랑주(Lagrangian)완화기법에 의한 분지한계법을 개발하였고, Hil and Benton[2]은 차량이동속도를 고려한 연구를 수행하여 절약형모형과 시간에 종속적인 이동속도에 대한 알고리즘을 제안하였으나 정확성 문제를 해결하지 못하였다. Fisher and Jaikumar[8]는 수학적 프로그래밍에 기반을 둔 발견적 기법을 제안하였으며 Solomaon[10]은 Clarke and Wright[4]의 절약개념에 시간대의 제약조건을 적용하였다. 최적 조합문제를 다루는 다양한 형태의 탐색 알고리즘에 관한 연구는 Boese[6], Jones, Li and Rhinehart[5], Tang[12], Zabinsky[13] 등이 있다. Lin-Kernighan[7]은 일반적인 TSP에 대해 13,509개의 도시문제에 적용 가능한 실행결과가 좋은 휴리스틱 해법을 개발하였다.

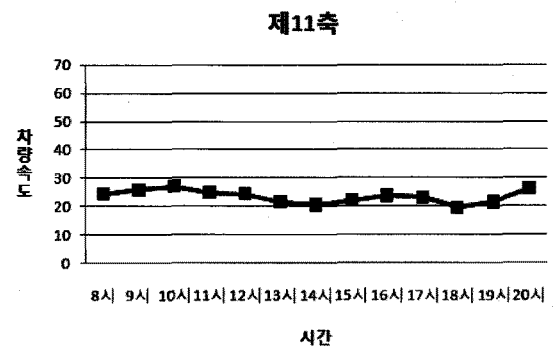
2.2 시간대에 따른 차량통행속도의 변화

B시의 조사 자료[1]는 도로 유형별 통행속도, 권역별

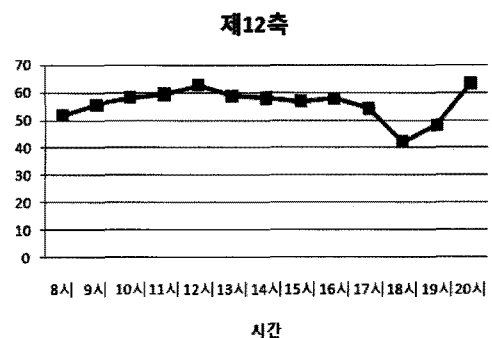
최저·최고 통행속도와 주요 도로구간의 시간대에 따른 속도 비교로 나누어져 있다. 도로 유형별 통행속도는 B시를 권역 즉 도심, 외곽, 도시고속도로로 나누어서 평일 차량 평균통행속도를 조사한 결과로써 도심이 외곽에 비해 정체가 많이 일어나며 도시고속도로의 차량통행속도가 도심이나 외곽도로에 비해 2~3배 빠르다는 것을 보여주었다. 이 자료를 분석한 결과 다음과 같은 사항들을 발견할 수 있었다.

무엇보다 먼저 특정도로를 이용하면 임의의 목적지까지 2~3배 빠른 속도로 이동할 수 있다는 점이다. 그리고 특정시간대에 차량통행속도가 높은 구간을 선택해서 차량경로를 형성하는 것이 운행시간을 줄이고, 배송량을 증가시킬 수 있다. 그러므로 현실 적용성 제고를 위해 각 도로의 시간대별 차량통행속도를 고려한 차량경로 탐색 해법이 개발되어야 한다.

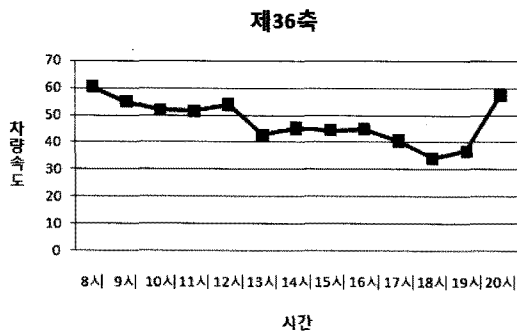
B시의 경우 총 36개의 교통축별 차량 속도자료가 시간대별로 조사되어 제공되는데, 시간대별 속도변화를 분석한 결과 대표적인 형태 3가지를 나타낸 그래프를 <그림 1>에서 <그림 3>에 나타내었다. 이 그래프들은 시간대에 따라 변하는 차량의 통행속도를 보여주고 있다. 이 그래프들을 보면 지역과 시간구간에 따라 차량 통행 속도변화에 많은 차이가 있음을 보여주고 있다.



<그림 1> 11축의 시간대별 차량통행속도



<그림 2> 12축의 시간대별 차량통행속도



〈그림 3〉 36축의 시간대별 차량통행속도

2.3 차량통행속도의 특성 및 시간구간의 분할

차량통행속도의 차이는 시간대와 지역에 따른 정체 현상에 의한 것이며 이러한 정체현상은 평일인 경우 거의 일정한 시간대에 같은 정체현상이 나타난다. 또 시간대에 따라 차량통행속도 변화가 거의 없는 구간이 있는 반면 변화가 매우 큰 구간도 있다. 도시마다 약간의 차이는 있겠으나 B시의 경우 하루 종일 큰 변화가 없는 지역을 제외하고, 가장 빠른 속도와 느린 속도의 차이가 20% 이상 나는 지역들의 시간별 차량속도를 분석하면 공통적으로 차이가 나는 5개의 시간대 구간으로 나눌 수 있다.

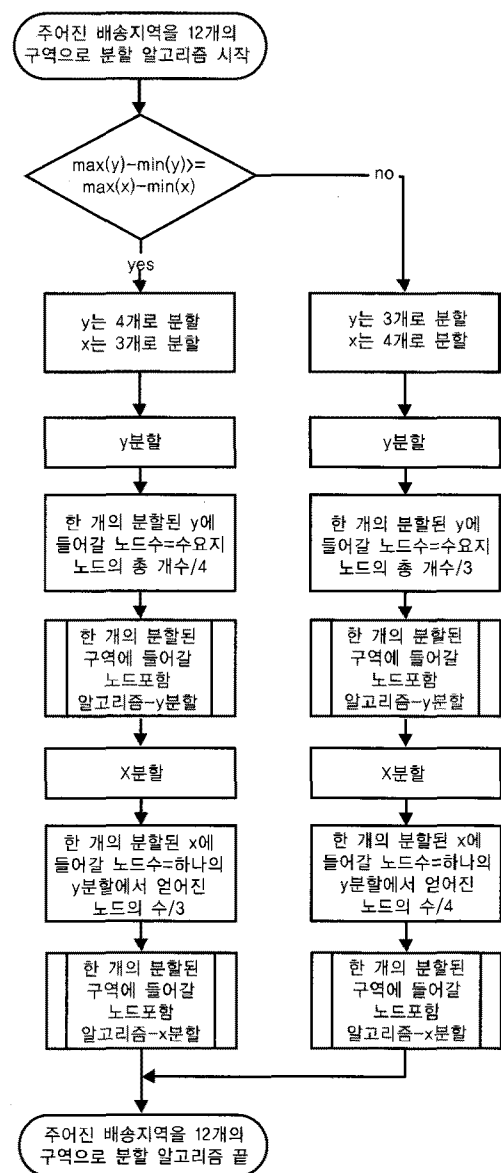
1번 시간대는 8:00시~10:00시의 2시간이며, 차량통행 속도는 최저 10km/h, 최고 55km/h로 구간에 따라 차량 속도가 45km/h의 차이를 보인다. 2번 시간대는 10:00시~13:00시의 3시간으로 20km/h가 최저속도이며 최고속도는 60km/h이다. 3번 시간대는 4시간, 4번 시간대는 2시간, 5번 시간대는 1시간으로 나누어지며, 총 12시간이다. 이렇게 5가지의 시간구간대로 나누면, 각 구간 내에서는 차량의 속도가 변화가 없는 동일한 것이 되므로 일반적인 TSP(traveling salesman problem)가 되어 최단경로를 찾는 문제의 해법이 간단하게 된다.

3. 일반 TSP로 변환을 위한 구역분할 해법의 설계

3.1 시간구간 및 구역분할 절차

먼저 정해진 배송시간이 주어지고 거점과 수요지의 위치에 따라 전체 배송구역이 정해진다고 가정한다. 배송시간은 12시간이고 이 시간을 시간대에 따른 속도 변화에 의거 5개의 시간구간대로 나눈다. 둘째, 배송구역을 12개의 구역으로 분할한다. 12개로 분할하는 절차는

〈그림 4〉와 같다. 셋째, 주어진 시간의 다음 시간대의 수요구간사이의 이동시간과 주어진 시간대에서 수요구간 이동시간의 차이를 구한다. 넷째, 차량의 현재위치에서 각 구역의 중심점까지의 거리를 계산한다. 다섯째, 각 구역의 수요지의 이동시간 차이 합에서 차량의 현재위치에서 각 구역의 중심점까지의 이동시간을 뺀다. 이 값이 가장 큰 것을 배송할 구역으로 정한다. 여섯째, 정해진 시간대에 적당한 시간간격의 수만큼 배송할 구역에 인접한 구역을 할당한다. 일곱째, 다음시간대의 배송구역을 결정하기위해 셋째 단계로 돌아가서 같은 방법으로 다음 배송구역을 결정한다. 이와 같은 방법으로 5개의 시간대에 적당한 순서대로 정해진 구역 5개를 결정한다.



〈그림 4〉 12개 구역의 분할 절차

다음은 결정된 배송구역의 순서에 따라 초기차량 경로를 다음과 같이 구한다. 각 구역사이의 수요지 차량 경로는 운행시간이 가장 짧은 것으로 결정한다. 그 다음 하나의 구역 안에서는 결정된 수요지 노드를 제외한 나머지 노드에 대해 최단 거리법으로 경로를 결정한다. 초기해의 경로에 의해 운행시간을 계산하는데, 각 시간대에 포함되어야 하는 노드가 실제로 그 운행시간대내에 포함되어지는지 확인하여 각 시간대의 운행시간을 초과하는 노드에 대해서는 그 다음 시간대의 운행시간으로 재계산한다. 다음은 구역분할을 위해 배송구역을 총 배송시간으로 나눈다. 여기에서는 배송시간이 12시간이므로 배송구역을 12개로 나누는데 구역분할 절차를 모두 종료하면 <그림 5>와 같이 구역분할이 완료된다.

3.2 배송구역의 결정

차량통행속도는 각각의 수요지 노드간의 경로에 대해 차량통행속도를 평균한 값으로써 'km/h'로 나타낸다. 이 속도(km/h)와 각 수요지 노드간의 거리(km)는 주어져 있고 이것으로 수요지 노드 간 운행시간을 계산한다. 운행시간(h) = 거리(km) / 속도(km/h)이다. 각 시간대에 따른 운행시간이 계산되었으면 '시간대에 따른 차량운행시간 차이'를 계산하여 배송구역의 순서를 결정한다. 계산과정을 구체적으로 표현하면 다음과 같다. 여기에서 '시간대에 따른 차량운행시간 차이'는 '차이(h)'로 표기한다.

- [단계 1] 각 노드에서 주어진 시간대의 다음 시간대와 주어진 시간대의 차량통행속도를 사용하여 '차이(h)'를 계산한다.
- [단계 2] 각 노드에 대해 모든 구간의 '차이(h)'를 합산한다.
- [단계 3] 구역분할 절차에서 12개로 분할한 각 구역에 대해 모든 노드들의 '차이(h)'를 합산한다. 각 구역에 속해있는 모든 노드의 '차이(h)'를 합산한 것을 이후로 ' $\sum_{i=0}^N$ 차이(h)'이라고 표기한다(N : 하나의 구역에 포함되어 있는 노드의 수).
- [단계 4] '각 구역까지의 운행시간'을 계산한다. 계산을 위해 '시작점'은 1시간대에서는 거점을 사용하며, 2, 3, 4, 5시간대에서는 앞 단계에서 계산된 점을 사용한다. '각 구역까지의 운행시간'은 '시작점'에서부터 각 구역의 중심점까지의 직선거리를 주어진 시간대의 평균 차량통행속도로 환산하여 운행시간으로 계산한 것이다. '각 구역까지의 운행시간'은 '시작점'에서 각 구역의

중심점까지의 직선거리(km)를 현재 시간대의 평균 차량통행속도로 나누어서 계산한다. '시작점'의 각 구역의 중심점 두 점이 주어졌을 때 이 두 점의 직선거리는 $\sqrt{(x'-x)^2+(y'-y)^2}$ 이다. 그리고 이 직선거리를 주어진 시간대의 평균 차량통행속도로 나누면 주어진 시간대에서의 '각 구역까지의 운행시간'이 계산된다.

- [단계 5] '주어진 시간대 구역선택 이익 시간'이 가장 큰 구역을 배송할 구역으로 선택한다. 이 시간대는 다음과 같이 계산하며, 12개의 구역 중에서 이 값이 가장 큰 구역을 '시작점'에서 다음 배송구역으로 선택한다.

$$\text{'주어진 시간대 구역선택 이익 시간'} = \sum_{i=0}^N \text{차이}(h) \cdot \text{'각 구역까지의 운행시간}(h)$$

3.3 인접구역 및 배송순서의 결정방법

인접구역을 결정하기 위해서는 먼저 검토대상구역이 인접한 구역인지의 확인과정이 필요하다. 이것은 앞에서 구역을 분할했던 12개의 구역분할에서 나온 구역의 모서리점이 사용된다. '주어진 시간대 구역선택 이익 시간'이 가장 큰 하나의 구역이 결정되었다면 결정된 구역의 4개의 모서리 점을 포함하는 선분이 하나라도 있는 구역은 결정된 구역의 인접한 구역이 된다.

- [단계 1] 다음의 입력 값을 '구역분할절차'에서 받는다.
 x_1, x_2, y_1, y_2 : 전체 배송영역의 최소값 x_1, y_1 과 최대값 x_2, y_2 .
 $\max x, \min x$: 가변이동속도하의 '배송구역 결정절차'에서 선택된 구역의 최대 x값과 최소 x값.
- [단계 2] x_2-x_1 과 y_2-y_1 중 어느 것이 큰지 확인한다. x, y축으로 몇 개로 나뉘었는지 확인한다.
- [단계 3] 앞쪽 계산절차를 통해 '현재 시간대 구역선택 이익 시간'으로 결정된 구역번호를 확인한다.
- [단계 4], [단계 3]의 선택된 구역번호를 중심으로 인접해 있는 구역을 확인한다.
 - ① 선택된 구역의 옆 위치 확인
 - ② 선택된 구역의 위치 옆에 따라 좌, 우 구역 선택
 - ③ 선택된 구역의 위치 행에 따라 상, 하 구역 선택 모서리 값 $\max x$ 와 $\min x$ 값을 포함하는 상, 하 구역선택
- [단계 5] 인접된 구역의 번호를 저장한다.

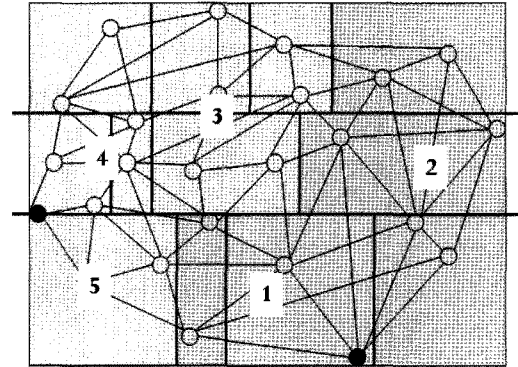
인접구역 결정 방법은 배송구역 결정 방법에서 선택된 구역과 동일 시간대에 배송하기에 적합한 구역을 결정하는 것이다. 인접구역 결정 단계에서는 인접한 구역들 중 현재 배송하는 것이 가장 유리한 '주어진 시간대 배달 이익 시간'이 큰 것을 시간대에 필요한 구역만큼 포함시킨다. '주어진 시간대 구역선택 이익 시간'을 계산할 때는 구역분할 절차를 통해 분할된 구역, 중심점과 모서리 점, 배송순서결정 절차의 구역에서 계산된 $\sum_{i=0}^N$ 차이(h)'와 선택된 구역중심에서 각 구역중심점까지의 거리를 이용하여 앞서 선택된 구역에 인접시킬 구역을 결정한다.

선택된 구역에 인접시킬 구역을 결정할 때는 시간대에 맞도록 구역의 수가 결정되어야 하는데, 가변이동속도 하의 배송구역 순서 결정 절차에서 결정된 배송구역을 기준으로 1번 시간대는 2시간이므로 구역을 2개, 2번 시간대에는 3시간이므로 구역을 3개, 3번 시간대는 4시간이므로 4개 구역, 4번 시간대에는 2시간이므로 2개 구역 그리고 5번 시간대에는 1시간이므로 1개의 구역을 배정한다. 인접구역 결정순서는 다음 각 단계를 따른다.

- [단계 1] 인접구역확인절차를 통하여 인접한 구역들을 탐색한다.
- [단계 2] 인접한 구역 중 배송구역으로 결정되지 않은 구역들을 탐색한다. [단계 1]과 [단계 2]를 거치면 인접 구역이 결정된다.
- [단계 3] 인접구역 중에 배송구역으로 결정되지 않은 구역들의 각각의 '차이(h)'를 계산한다.
- [단계 4] 가장 큰 $\sum_{i=0}^N$ 차이(h)'를 가진 구역이 인접구역으로 결정되어 하나의 시간대에 배송할 구역으로 결정된다.
- [단계 5] 하나의 시간대에 할당할 배송구역의 수가 남았으면 [단계 3]으로 가서 이 절차를 반복한다.
- [단계 6] 배송인접 구역이 정해졌으면 정해진 구역을 제외한 구역에서 배송구역결정절차를 이용하여 다음 시간대에서의 배송구역을 결정한다. 모든 구역의 순서가 결정될 때까지 배송구역 결정절차, 인접구역확인절차, 인접구역결정 절차를 반복한다.

<그림 5>는 모든 구역의 순서가 결정된 상태이다. 1번 시간대에 2구역, 2번 시간대에 3구역, 3번시간대에 4구역, 4번시간대에 2구역, 5번시간대에 1개 구역이 할당되었다. 배송의 순서는 시간대 1의 배송 구역에서 시간대 2의

배송 구역, 시간대 3의 배송 구역, 시간대 4의 배송 구역, 시간대 5의 배송구역 순서로 배송하게 된다.



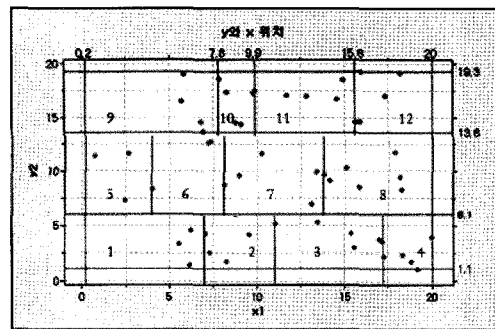
<그림 5> 배송구역 순서 결정

4. 구역분할해법의 적용

4.1 60개 수요지 위치정보와 12개 구역분할

60개의 수치 예에서 사용할 x, y좌표는 임의로 발생시키며 <그림 6>에 좌표로 표시되어있다. 전체 배송구역은 전체 배송시간인 12시간에 따라 12개의 배송구역으로 분할한다. 앞 절의 자료 분석의 결과에 따라 12시간은 5개의 시간대 구간으로 나누고 나누어진 5개의 시간대에 알맞게 12개의 구역을 결정한다.

전체 구역은 <그림 6>에서와 같이 수요지 노드 60개를 각각 20개씩 포함하도록 3등분된다. 3등분된 구역 중 아래구역의 노드 20개의 위치 좌표를 기준으로 다시 4등분한 결과이며 왼쪽아래구역부터 1번 구역, 2번 구역, 3번 구역, 4번구역이라고 한다. 4등분한 하나의 구역 안에 들어가는 노드 수는 각각 5개이다. <그림 6>에는 12개로 분할된 각 구역에 포함된 노드에 대한 정보가 나와 있다.



<그림 6> 12개의 분할구역과 수요지 노드

4.2 배송구역별 구역 ‘차이(h)’ 값의 합 계산

각 구역의 ‘차이(h)’값은 주어진 시간대의 다음 시간대에서 그 구역에 포함된 노드 중 어떤 노드에서 다른 수요지 노드로 가는 운행시간에 주어진 시간대에서 앞의 같은 노드 간의 운행시간을 뺀 것이다. 60개 수요지 노드의 각 시간대에서 각 수요지 노드간의 운행거리는 60개 수요지 노드의 위치정보에 의해 계산되며, 차량통행속도의 예시는 <표 1>에 있다.

각 시간대의 수요지 노드간의 운행시간의 ‘차이(h)’를 계산하며 이 ‘차이(h)’값은 <표 2>에 나와 있다. <표 2>에 나와 있는 각 노드의 ‘차이’는 각 노드에서 운행가능한 모든 경로에 대한 각 시간대의 ‘차이(h)’값을 모두 더한 값이다.

<표 1> 시간대1에서 노드간의 차량운행속도(km/h)

노드번호	0	1	2	3	4	5
0	0.00	23.55	16.74	19.33	24.51	18.00
1		0.00	21.27	25.41	53.65	25.30
2			0.00	36.13	52.45	22.00
3				0.00	10.46	44.46
4					0.00	39.51
5						0.00

<표 2> 각 수요지노드 간 운행시간의 차이

2번 시간대 - 1번 시간대		3번 시간대 - 2번 시간대		4번 시간대 - 3번 시간대		5번 시간대 - 4번 시간대	
노드	각 노드의 ‘차이’	노드	각 노드의 ‘차이’	노드	각 노드의 ‘차이’	노드	각 노드의 ‘차이’
0	-4.45	0	-2.00	0	15.33	0	-8.28
1	-6.07	1	-2.58	1	15.59	1	-3.93
2	-5.95	2	-3.07	2	25.35	2	-10.45
3	-7.03	3	-2.73	3	15.32	3	-3.55
4	-5.99	4	-2.60	4	16.24	4	-3.04
5	-4.33	5	-2.01	5	14.90	5	-6.63

4.3 배송구역별 중심점까지의 거리계산

배송구역순서 결정의 방법은 각 구역에 속한 노드들의 ‘차이(h)’값의 합을 계산하여 이 값이 가장 큰 구역을 선택하는 것이다. 그리고 시간대에 알맞은 개수가 몇 개인지 확인하고 시간대에 알맞은 개수만큼의 인접구역을 선택하여 배송구역을 결정한다.

거리 계산에 필요한 데이터는 각 구역의 모서리 점과 모서리 점에 의해 계산된 구역의 중심점 x, y 그리고 ‘시작점’ x, y이다. 60개 수요지 노드의 수치 예에서 각 구역의 모서리 점과 중심점은 <표 3>과 같다. 시간대 1에서 사용된 ‘시작점’ x, y는 거점이다.

<표 3> 12개 구역의 모서리 점과 중심점

구역번호	모서리 점	왼쪽 아래	왼쪽 위	오른쪽 아래	오른쪽 위	중심점
1번 구역	x	0.2	0.2	7	7	3.6
	y	1.1	6.1	1.1	6.1	3.6
2번 구역	x	7	7	11	11	9
	y	1.1	6.1	1.1	6.1	3.6
3번 구역	x	11	11	17.2	17.2	14.1
	y	1.1	6.1	1.1	6.1	3.6

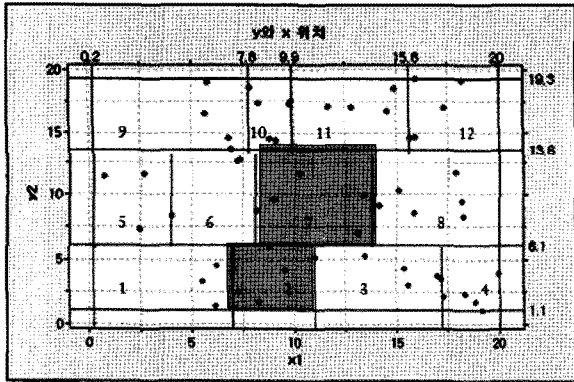
4.4 배송구역에서 인접구역의 결정

특정 시간대에서 12개의 구역 중 ‘차이(h)’값이 가장 큰 구역이 계산에 의해 결정되면 정해진 구역의 인접구역을 결정한다. 가장 큰 ‘차이(h)’값으로 7번구역이 시간대 1의 배송구역으로 결정되었다. 그 다음에 7번구역과 인접한 구역 중에 현재 시간대 즉 시간대 1에 배송하면 좋은 구역들을 결정하게 된다.

<표 4> 인접구역결정 관련 값의 예시(2번-1번 시간대에 대한 7번 구역)

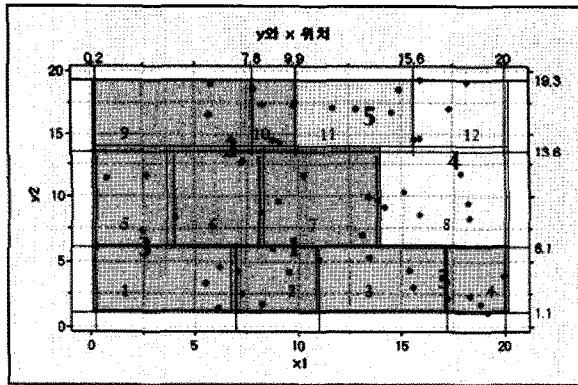
두 구역 번호	7번, 2번 구역		7번, 3번 구역		7번, 4번 구역		7번, 8번 구역		7번, 10번 구역		7번, 11번 구역	
(a)두 구역의 차이의 합	-45.06		-53.46		-45.12		-49.97		-48.51		-53.07	
두 구역 중점(x, y)	8.6	7.9	8.6	10.4	8.6	12.7	11.7	11.8	15	7.8	15	9.8
두 구역 중점까지의 거리	5.08		6.17		7.72		5.39		1.8		3.33	
(b)거리에 의한 운행시간 (h)	1.39		1.44		2.37		1.94		0		0	
(a)-(b) (h)	-46.45		-54.9		-47.49		-51.91		-48.51		-53.07	
거점부위의 거리	10.16		12.33		15.44		5.93		3.73		5.56	

<표 4>의 자료를 보면 7번과 2번 구역의 두 구역의 '차이(h)합 - '거리에 의한 운행시간'이 가장 큰 -46.45 시간이므로 7번 구역의 인접구역으로 2번 구역이 결정된다. 이의 결과는 <그림 7>에 나타나있다.



<그림 7> 시간대 1의 배송구역 결정 예시

시간대 1에서 시간대 5까지의 배송구역의 순서는 <그림 8>과 같다.



<그림 8> 5개의 시간대에 따른 배송구역 순서 결정

결정된 배송구역 순서는 시간대 1에는 2번, 7번 구역, 시간대 2에는 6번, 9번, 10번 구역, 시간대 3은 1번, 5번, 3번, 4번 구역, 시간대 4에는 8번, 12번 구역, 시간대 5는 11번 구역이 결정되었다.

시간대에 따라 배송구역의 순서가 결정되면 시간대에 따라 결정된 구역안의 노드들은 그 시간대에 배송되는 노드로 결정되고 하나의 시간대 안에 포함된 노드들의 배송순서도 결정해야 한다.

4.5 배송구역 순서 결정 후 수요지별 방문순서 결정

이상과 같은 절차에 따라 12개의 배송구역이 차량속도변화가 없는 5개의 배송구역으로 묶어졌으며, 5개의

배송구역별로 배송해야할 순서가 결정되었다. 이 5개의 배송구역은 배송구역별로 차량의 이동속도 변화가 없이 일정한 구역이 되었으므로 현재까지 많이 개발되어있는 일반적인 TSP해법의 적용이 가능해졌다.

5개의 각 배송구역별로 출발점과 끝점이 있으며, 출발점으로 돌아오지 않는 최단경로 탐색을 한 후, 이 경로들을 5개의 배송구역 전체에 대하여 연결해 주면 큰 하나의 경로가 완성되게 된다. 여기에서 각 배송구역별로 출발점과 끝점의 선택이 중요한데, 다음 배송구역의 출발점과 연결되어야하므로 다음의 순서에 따라 이 과정을 수행한다.

먼저 시간대 1에 배송할 구역으로 결정된 구역의 노드들 중에서 거점부터 이들 노드들 사이의 운행시간이 가장 짧은 노드를 처음 방문할 노드로 한다. 그리고 시간대 1에 배송할 구역의 마지막 노드와 시간대 2에 배송할 구역의 연결할 노드를 결정한다. 시간대 1에 배송할 구역에서 첫 노드로 결정된 노드를 제외한 모든 노드와 시간대 2에 배송할 구역의 모든 노드들의 운행시간을 비교하여 가장 짧은 운행시간을 가지는 두 노드를 선택한다. 그러나 시간대 1의 배송할 구역에서 시간대 2의 배송할 구역으로 넘어갈 때에는 운행시간이 변경되므로 이들 노드 사이의 운행시간은 시간대 1과 시간대 2의 운행시간 평균으로 계산한다. 이와 같은 방법으로 시간대 5의 배송 구역까지 시간대에 의한 배송구역사이를 연결할 노드들을 결정한다. 시간대 5의 배송 구역에서 마지막 노드를 결정할 때에는 거점 노드로 돌아가야 하므로 시간대 5의 배송 구역에 속한 노드들 중 연결노드로 선택된 노드를 제외하고 거점 노드와의 운행시간이 가장 짧은 노드를 선택한다.

5. 결 론

도시에서의 배송차량 최적경로 결정문제는 일반적인 TSP처럼 최단경로를 찾는 문제와는 달리 1시간 간격으로 차량의 이동속도가 변경되는 매우 복잡한 문제이기 때문에 효율적이 해법의 필요성이 매우 높은 상황이라 하겠다. 증가하는 택배업에서의 수요만큼, 해법의필요성은 높으나 문제의 복잡성으로 효율적인 해법의 개발은 간단한 문제가 아니다. 본 연구에서는 이러한 문제에 대한 해법으로 차량의 속도변화가 없는 배송구역의 결정방법과 나뉘어진 배송구역으로 배송순서를 결정하는 해법을 설계 제시하였다.

12개의 다른 속도를 가진 시간대를 5개로 단순화하여 배송구역을 설정함으로써 그 내부에서는 단일의 차량 이동 속도를 보이게 되어 출발점으로 돌아오지 않는

일반적인 TSP로 문제를 변환함으로써 효율적인 경로의 탐색이 가능하도록 하였다. 이 연구는 B시의 실제 교통자료를 분석하여 5개로 분할한 것이나, 정부의 차량 이동속도 조사가 시행되고 그 정보가 제공되는 도시에서는 어디에서 그 자료를 분석하여 5개 혹은 기타 개수로의 차량이동속도에 변화가 없는 구역의 개수로 분할하여 제시한 해법절차의 적용이 가능할 것이다.

추후 연구과제로 1차 연결된 전체의 경로에 대해 1 시간간격의 실제 차량속도정보에 따라 2-opt 혹은 3-opt로 개선안을 찾은 방안의 구체적인 설계와 적용이 남아있다.

참고문헌

- [1] 부산광역시; “부산광역시 차량통행속도 조사결과”, 2008년도 부산광역시 교통조사, 부산시, 2009.
- [2] A. V. Hill and W. C. Benton; “Modelling intra-city time-dependent travel speeds for vehicle scheduling problem,” *Journal of the Operational Research Society*, 43(4): 343-351, 1992.
- [3] G. B. Dantzig and J. H. Ramser; “The truck dispatching problem,” *Management Science*, 6(1): 80-91, 1959.
- [4] G. Clarke and J. Wright; “Scheduling of vehicles from a central depot to a number of delivery point,” *Operations Research*, 2(4) : 568-581, 1964.
- [5] J. Y. Li and R. R. Rhinehart; “Heuristic random optimization,” *Computers and Chemical Engineering*, 22 : 427-444, 1998.
- [6] K. D. Boese, A. B. Kahng, and S. Muddu; “A new adaptive multi-start technique for combinatorial global optimizations,” *Operations Research Letters*, 16 : 101-113, 1994.
- [7] Keld Helsgaun; “An effective implementation of the Lin-Kernighan traveling salesman heuristic,” *European Journal of Operational Research*, 126 : 106-130, 2000.
- [8] M. Fisher, R. Jaikumar; “A generalized assignment heuristic for the vehicle routing problem,” *Networks*, 11 : 109-124, 1981.
- [9] M. G. Kang; “네트워크와 알고리즘”, 박영사, 1999.
- [10] M. Solomon; “Algorithms for the vehicle routing and scheduling with time window constraints,” *Operations Research*, 35: 254-265, 1987.
- [11] N. Christofides and S. Eillon; “An algorithm for the Vehicle Dispatching,” *Operations Research*, 20: 309-318, 1969.
- [12] Z. B. Tang; “Optimal sequential sampling policy of partitioned random search and its applications,” *Journal of Optimization Theory and Applications*, 102 : 157-175, 1997.
- [13] Z. B. Zabinsky, “Stochastic methods for practical global optimization,” *Journal of Global Optimization*, 13 : 433-444, 1998.