

An Application of k-Means Clustering to Vehicle Routing Problems

Je-Min Ha · Geeju Moon[†]

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

K-Means Clustering의 차량경로문제 적용연구

하제민 · 문기주[†]

동아대학교 산업경영공학과

This research is to develop a possible process to apply k-means clustering to an efficient vehicle routing process under time varying vehicle moving speeds. Time varying vehicle moving speeds are easy to find in metropolitan area. There is a big difference between the moving time requirements of two specific delivery points. Less delivery times are necessary if a delivery vehicle moves after or before rush hours. Various vehicle moving speeds make the efficient vehicle route search process extremely difficult to find even for near optimum routes due to the changes of required time between delivery points. Delivery area division is designed to simplify this complicated VRPs due to time various vehicle speeds. Certain divided area can be grouped into few adjacent divisions to assume that no vehicle speed change in each division. The vehicle speeds moving between two delivery points within this adjacent division can be assumed to be same. This indicates that it is possible to search optimum routes based upon the distance between two points as regular traveling salesman problems. This makes the complicated search process simple to attack since few local optimum routes can be found and then connects them to make a complete route. A possible method to divide area using k-means clustering is suggested and detailed examples are given with explanations in this paper. It is clear that the results obtained using the suggested process are more reasonable than other methods. The suggested area division process can be used to generate better area division promising improved vehicle route generations.

Keywords : Vehicle Routing Problem, k-Means Clustering, Traveling Salesman Problem

1. 서 론

차량경로문제(VRP : Vehicle Routing Problem)는 일반적으로 차고지 혹은 물류창고에서 한 대 또는 복수대의 차량이 다수의 고객에게 재화 및 서비스를 전달하고 다시 원점으로 돌아오는 최선의 경로를 결정하는 문제이다. 문제의 형태에는 차량 대수, 창고 수, 차량 용량 등의 제약 하에 최소의 비용으로 수요를 충족하는 경로 찾기

등이 있다. VRP는 택배회사나 화물운송 등 차량을 이용하는 기업에서 업무 효율 향상을 위해 활용될 수 있다.

본 연구는 하나의 창고에서 한 대의 차량이 각 배송지점을 방문한 후 다시 원점으로 돌아오는 경우의 문제를 대상모형으로 하며, 배송품의 무게, 부피 등의 제약은 배제된 경우를 다룬다. 이동시간대에 따라 배송지점들 사이에 차량의 이동속도가 변화하는 경우를 다룬 것이 본 연구의 주된 특징이다. 이동하는 시간대의 변화에 따라 이동속도가 바뀌는 경우 효율적인 배송경로 탐색이 매우 어려워진다. 이러한 어려움을 해결하기 위해 Moon과 Park [10]은 차량의 이동속도에 변화가 없는 시간대를 분석하

Received 6 August 2015; Finally Revised 26 August 2015;
Accepted 2 September 2015

[†] Corresponding Author : gjmoon@donga.ac.kr

였고, 변화가 없는 시간대로 구역 분할한 후 각 분할된 구역 내에서는 일반적인 TSP(Traveling Salesman Problem)로 지역적 경로를 구한 후 연결하여 전체 경로를 구성하는 탐색법을 제시하였다. 본 연구에서는 이동경로의 구성을 위한 구역의 분할에 있어서 보다 효율적으로 구역분할을 수행할 수 있는 해법을 설계하였다. 이전 연구에서의 구역 분할은 배송 지점들의 개수에 따른 단순 분할이었으나, 본 연구에서는 k-means clustering을 활용함으로써 보다 합리적인 분할이 가능하게 하였다. K-means clustering을 사용하여 구역을 나누면 기존의 단순한 개수 기준 분할법에 비해 합리적인 결과를 도출할 수 있게 된다.

본 논문의 구성을 간략히 설명하면 제 1장에는 서론, 제 2장에는 관련 문헌연구가 정리되어 있다. 제 3장에는 먼저 k-means clustering 알고리즘이 상세히 설명되어 있으며, 이를 활용하여 구역분할 하는 방법이 예와 함께 서술되어 있다. 제 4장에는 설계된 해법이 예제문제와 함께 구체적으로 설명되어 있고, 제 5장에는 결론이 기술되어 있다.

2. TDVRP와 관련 해법

일반적인 차량경로문제 연구의 대부분은 두 배송지점 간의 이동시간을 거리에 비례하는 고정된 값으로 가정하고 있다. 그러나 실제 운송이 이루어질 때의 이동소요시간은 고정된 값이 아닌 도로상황이나 환경적 요인에 영향을 받아 큰 변화를 보인다. 두 지점사이의 이동소요시간이 일정하다고 가정하는 것은 여러 이유로 이동소요시간이 달라지는 구간의 경우 문제가 발생하게 된다. 여기에 관련된 연구로 시간의 변화에 따라 이동시간이 변동하는 차량경로문제(TDVRP : Time Dependent Vehicle Routing Problem)를 들 수 있다. 이 연구는 Malandraki와 Daskin[9]에 의해 처음 소개되었다. Kenyon과 Morton[4]은 지정된 시간에 서비스 시간을 최소화할 확률을 최대화하는 모형을 제시하였다. Haghani와 Jung[2]은 복수대의 차량, 각기 다른 차량용량, 실제 서비스 요청시간 등 여러 가지 제약에 유전자 알고리즘을 이용하여 문제를 해결하고자 했다. Donati 등[1]은 시간변화에 따른 각 배송지점 사이의 속도에 따른 최단경로를 찾기 위해 multi ant colony system을 이용한 메타휴리스틱을 제시하였다. Kuo 등[6]은 시간에 따른 속도의 변화와 상품을 싣고 내리는 시간도 고려하여 차량 경로 문제를 수정된 타부서치 기법을 이용해 문제를 해결하고자 하였다. Jabali 등[3]은 FIFO 원칙을 만족하기 위해 실제 데이터를 기반으로 한 속도를 함수로 변환해 경로문제를 해결하였고, Kuo[5]가 시간에 따른 주행속도에 따라 최적화된 차량 할당 및 차량경로문제를 해결하기 위해 non-passing 타부서치 휴리스틱을 제안하였다. Moon과 Park[10]은 시간대에 따라 변화하는 속도를 고려하여

배송지점을 분할하여 각 분할된 구역에 시간대에 따른 통행속도를 적용하여 문제를 해결하고자 하였다.

본 연구는 Moon과 Park[10, 11]의 연구 중 구역분할에 관련된 것의 개선에 관한 것이다. 시간대에 따라 구역설정을 할 때 단순히 x, y축을 기준으로 각 구역 당 동일한 개수의 배송지점을 할당한 것을 개선하여 배송지점들의 근접도에 따라 단위시간당 다른 개수가 배정되도록 한 것이다. 개수를 동일하게 배정하는 구역분할은 현실의 다양한 배송환경에 적합하지 않다. 현실의 배송상황에서는 배송해야 할 배송지점이 항상 동일한 개수로 나뉘지 않기 때문에 실제의 상황에 맞지 않는 어려움이 있었다. 본 연구에서는 k-means clustering 알고리즘을 사용하여 보다 합리적이고 현실 적용 용이한 구역분할 방법을 제시하고자 하였다.

3. 구역분할 방법의 설계

3.1 K-Means Clustering 알고리즘

K-means clustering 알고리즘은 MacQueen[8]에 의해 제안된 군집화 분석 알고리즘이다. 시작은 Lloyd[7]가 펄스 부호변조(Pulse-code modulation)의 기법으로 제안하였으며 이후 MacQueen이 처음으로 “k-means”라는 용어를 사용하여 알고리즘을 제안하였다.

기본적인 k-means clustering은 주어진 데이터(x_1, x_2, \dots, x_n)를 n보다 작은 k개의 cluster $S(S = S_1, S_2, \dots, S_k)$ 로 묶는 알고리즘으로 각 cluster와 데이터와의 거리차이 평균의 제곱합을 최소화하는 것이다. 이 알고리즘의 목적함수는 다음과 같다.

$$\min S = \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

이때 μ_i 는 cluster S_i 의 평균이다.

알고리즘은 초기 k를 임의로 설정하여 시작한다. 그리고 다음의 두 단계를 반복한다.

[단계 1] cluster 설정 : 각 점에 대해 그 점에서 가장 가까운 cluster를 찾아 배당한다.

임의의 값으로 cluster를 설정하고 각 데이터와 cluster 사이의 거리를 계산한다. 이때 사이의 거리계산은 유클리드 거리공식을 이용한다. 그리고 각 cluster를 기준으로 가까운 거리의 데이터로 군집을 형성한다.

[단계 2] cluster 재조정 : k를 각 cluster에 있는 점들의 평균값으로 재 설정해준다.

각 군집의 데이터의 속성평균을 이용하여 cluster의 값을 재설정한다. 해당 군집의 데이터 간 거리의 평균으로 cluster의 위치를 이동한다.

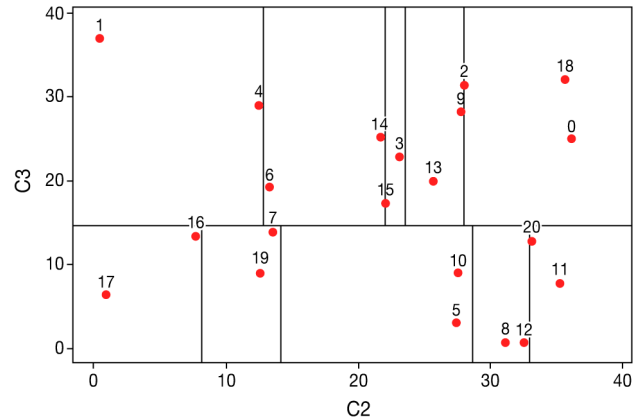
위의 두 단계를 계속 반복하여 cluster를 이동시키고 다시 군집을 형성하여 계산한다. 만약 계속된 계산 이후 cluster가 변하지 않는다면 반복을 종료한다.

3.2 배송구역의 분할

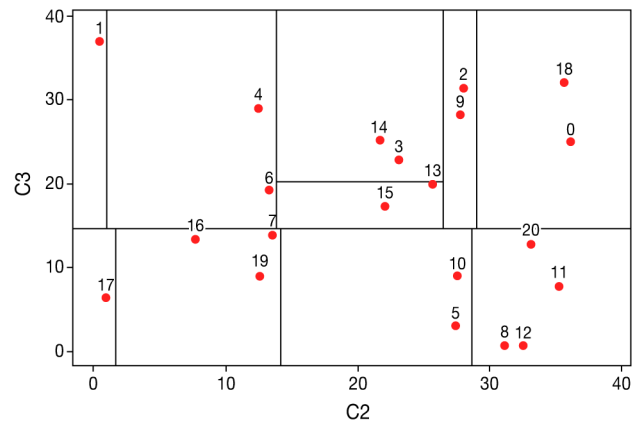
현실의 문제에서는 배송차량의 이동하는 시간대에 따라 차량속도가 달라지고 있으며, 이 경우 최적해를 찾기가 매우 어려워진다. 이 문제를 일반적인 TSP로 단순화하기 위해서는 차량의 이동속도에 변화가 없는 몇 개의 구역으로 대상지를 분할하는 것이 필요하다. K-means clustering으로 구역을 분할하기 위해서는 총 구역의 개수를 정하고 이 개수와 동일하게 k값을 정한다. 이는 k의 수가 분할될 구역의 수이기 때문이다. 본 연구에서는 Moon과 Park[11]에서와 같이 배송시간을 오전 10시부터 오후 8시까지 총 10시간으로 설정하여 10개의 구역으로 나눈다. 따라서 구역의 수와 동일하게 k는 총 10개이다. 10개로 나누어진 각 구역은 한 시간 동안 배송하게 될 구역을 표시하게 된다. 다음은 k-means clustering을 통해 분할된 구역에 각 시간대의 개수에 맞춰 인접구역을 정하고 각각의 시간대에 속도 값을 부여하여 차량이동시간을 계산한다.

<Figure 1>과 같이 이전 연구에서 x, y축을 기준으로 개수에 맞춰 구역분할을 했을 때는 각 구역에 모두 동일한 개수의 배송지점들이 할당되었다. 예제로 다루어 볼 문제의 총 배송지점 수는 20개이며 각 구역에 할당되는 배송지점의 개수는 두 개씩으로 모든 구역에 일정하다. 이는 총 배송지점의 수에 구역의 수를 나누어 할당되는 배송지점의 수를 계산한 것이다. 그러나 이 방법은 실제의 배송상황에는 적합하지 않음을 알 수 있다. 현실에서는 인접해 있는 배송지점들은 이동시간이 짧아서 동일한 시간에 여러 개를 배송할 수 있는 반면, 멀리 떨어져있는 지점들의 경우에는 이동시간이 길어서 배송 가능 개수가 줄어들기 때문이다. 이에 k-means clustering 알고리즘을 이용해 배송구역을 분할하면 이러한 단점을 보완할 수 있다. <Figure 2>는 <Figure 1>과 같은 배송지점들을 k-means clustering을 이용하여 구역을 나눈 결과이다.

<Figure 1>과 <Figure 2>를 비교해 보면 <Figure 2>에서는 각 구역에 다양한 수의 배송지점들이 할당된 것을 알 수 있다. 구역 당 1개에서 4개까지의 배송지점들이 할당되어 현실의 배송상황이 잘 반영되었다고 할 수 있다. 각 구역별로 동일한 시간동안 배송할 수 있는 개수가 배정되는 것이므로 모여 있는 것들은 많이 배정하고 멀리 떨어져있는 것은 이동시간이 많이 소요되므로 적은 개수로 배정되어야 하기 때문이다.



<Figure 1> Area Division by Moon and Park

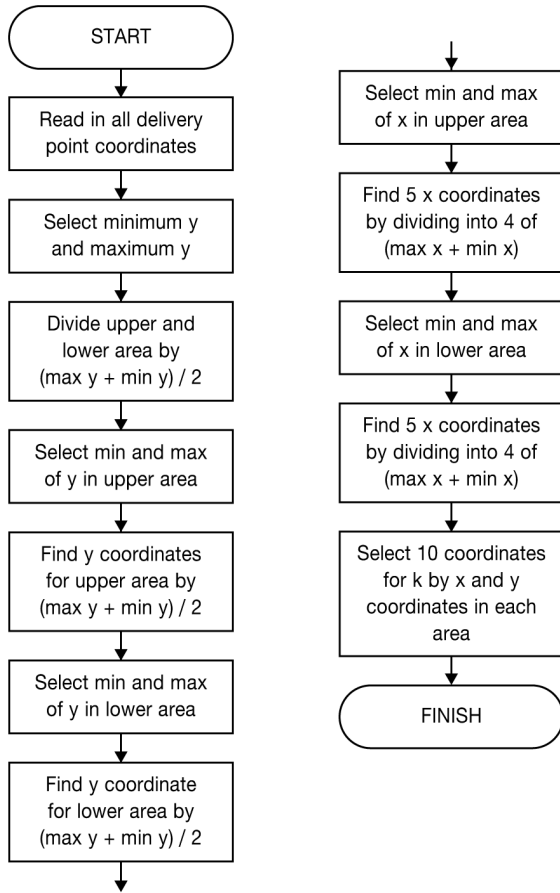


<Figure 2> Area Division by k-Means Clustering

4. 해법의 설계

4.1 초기 k위치의 결정방법

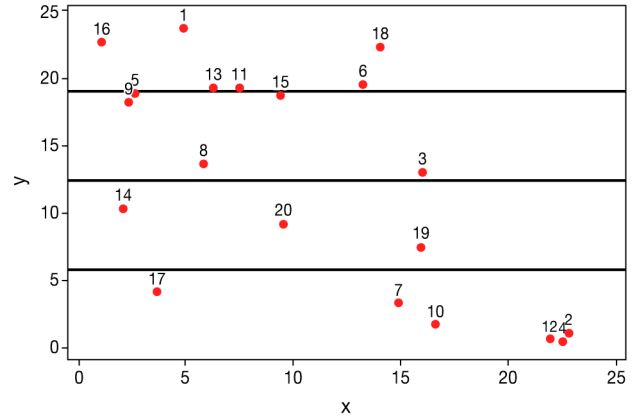
K-means clustering 관련 기존 연구들에서는 초기 k위치가 임의로 지정된다. 이는 임의로 결정되는 k의 위치에 따라 다양한 구역분할 값이 나올 수 있다는 것을 의미하는데, 만약 임의의 k의 위치가 어느 한 곳으로 편중된다면 그만큼 그룹화 되는 배송지점도 편중된다는 것이다. 그룹이 특정위치로 편중되면 그룹 간 거리의 차가 크게 발생할 수 있기 때문에 좋은 해를 구하기 어려워진다. 그룹 간 거리의 차가 크거나 짧으면 해법의 설계 시 여러 문제에 대한 활용성이 낮아지기 때문에 그룹 간 거리를 일정하게 지정해 주는 것이 바람직하다. 본 연구에서는 초기 k의 위치를 일정한 간격으로 고정시켜 구역분할의 기준점으로 지정한다. 초기 k위치 설정 방법은 <Figure 3>에 순서도로 표시되어있으며, 단계별 과정으로 표현하면 다음과 같다.



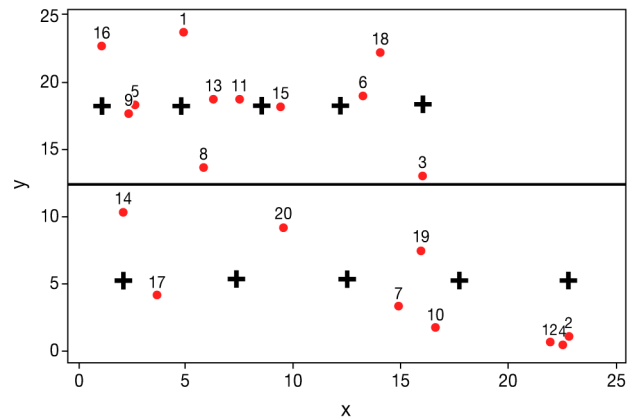
<Figure 3> Process for the Initial k Positions

- [단계 1] 배송지점의 x, y값(위도, 경도)을 읽어 들인다.
- [단계 2] y축에서 가장 큰 y와 가장 작은 y를 찾고 그 중간 값으로 나누어 윗구역과 아래구역으로 나눈다.
- [단계 3] 단계 2에서 나눈 윗구역과 아래구역의 가장 큰 y와 가장 작은 y를 각각 찾고 그 중간 값으로 <Figure 4>와 같이 각 구역의 y축으로 정한다.
- [단계 4] 단계 2에서 나눈 윗구역과 아래구역의 가장 큰 x와 가장 작은 x를 각각 찾고 그 중간 값을 4등분하는 x축 5개를 찾는다. max(x)와 min(x)의 더한 값을 4로 나누어 각 구역 x축의 간격을 찾고 max(x)와 min(x)를 포함하여 찾은 간격으로 5개의 x축을 찾는다.

<Figure 5>의 아래구역으로 예를 들면 min(x)의 값 2.14와 max(x)의 값 22.83의 합을 4로 나눈 값, 즉 $(22.83 + 2.14) / 4$ 의 값인 5.1725가 아래구역 x축의 간격이 된다. 그리고 max(x)와 min(x)를 포함하는 일정한 간격의 x축은 min(x)인 2.14에 찾은 간격 값인 5.1725를 더한 값 7.3125가 x축이 되는 것이다. 같은 방법으로 각 x축 12.485와 17.6575를 찾을 수 있다.



<Figure 4> Y Axes for Each Division



<Figure 5> Initial k Positions

- [단계 5] 단계 3과 단계 4에서 찾은 y축과 x축으로 각 k 점 10개를 찾는다. 결과는 <Figure 5>와 같다. <Figure 5>의 예에서 각 좌표의 위치는 아래 구역부터 오른쪽으로 $k_1 = (2.14, 5.425)$, $k_2 = (7.3125, 5.425)$, $k_3 = (12.485, 5.425)$, $k_4 = (17.6575, 5.425)$, $k_5 = (22.83, 5.425)$, $k_6 = (1.09, 18.445)$, $k_7 = (4.825, 18.445)$, $k_8 = (8.56, 18.445)$, $k_9 = (12.295, 18.445)$, $k_{10} = (16.03, 18.445)$ 이다.

이와 같이 초기 k의 위치를 일정한 간격으로 정해줌으로써 배송지점이 어느 한 위치로 편중되는 것을 예방할 수 있고, 또 전체 범위를 고려한 구역분할이 가능해질 수 있다. 여기에서 제시한 방법으로 한 시간에 배송할 배송지점들을 포함한 구역을 분할한 후 다시 차량의 이동속도 변화가 없는 것으로 가정된 4개의 시간구간대로 분할된 구역들을 묶어주면 배송경로 탐색에 활용이 가능해진다.

4.2 K-Means Clustering에 의한 구역분할

초기 k위치 설정이 끝났으면 k-means clustering 알고

리즘으로 구역분할을 시행한다. K의 개수는 본 연구에서 정의한 택배차량의 운행시간인 10시간을 기준으로 총 10개로 정한다. K-means clustering 알고리즘은 <Figure 6>의 순서도와 같이 시행하며 배송구역을 k의 개수인 10개로 나눈다. 다음은 알고리즘의 단계별 설명이다.

[단계 1] 이전 알고리즘인 초기 k위치 결정에서 결정된 k의 좌표를 찾는다.

[단계 2] 각 k와 배송지점간의 직선거리를 계산한다. 이때 직선거리는 유클리드 거리공식을 이용한다. 유클리드 거리공식은 다음과 같다.

$$d = \sqrt{(x' - x)^2 + (y' - y)^2}$$

d는 k와 배송지점 사이의 거리, x'와 y'는 k의 좌표, x와 y는 배송지점의 좌표, 단위는 km이다.

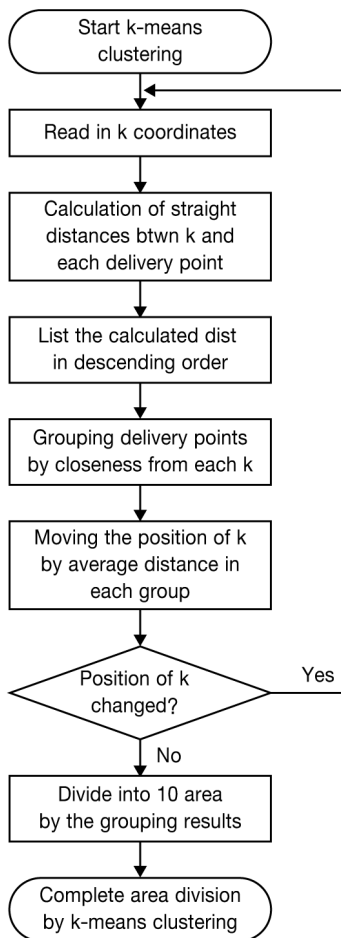
[단계 3] 계산한 직선거리와 k가 가까운 순서로 묶는다. 각 k와 각 배송지점의 거리가 근접 순으로 배정

하여 그룹화하는 것이다. <Figure 7>에 그룹화 결과가 나와 있다. <Figure 7>에서 '+'는 초기 k의 위치이며 각 k와 가까운 거리의 배송지점으로 그룹화 되었다.

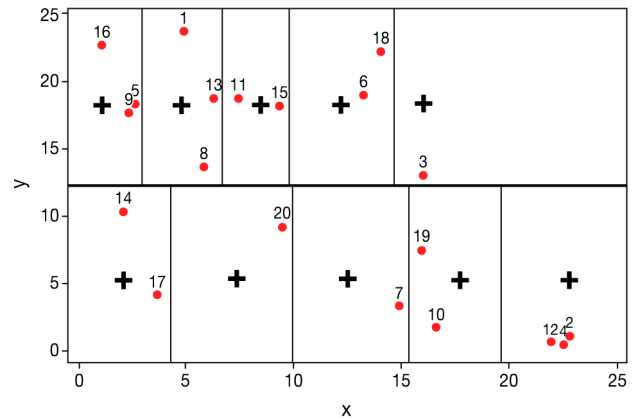
[단계 4] [단계 3]에서 그룹화 된 배송지점 간 평균을 계산하여 그 값으로 k의 위치를 재설정한다.

[단계 5] [단계 4]에서 k의 위치가 변경됐다면 다시 [단계 2]부터 각 과정을 반복한다. K의 위치가 변경되면 가까운 배송지점의 정보도 바뀌기 때문에 배송지점과의 거리를 재계산하여 다시 그룹화 한다.

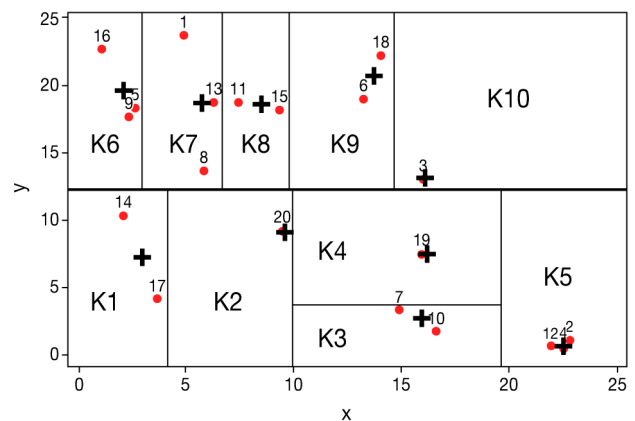
[단계 6] 이후 k의 위치에 변경이 없으면 반복과정을 종료하고 그룹을 결정한다. 그룹화한 결과는 <Figure 8>과 같다. <Figure 7>과 <Figure 8>을 비교하면 k의 위치가 변경된 것을 알 수 있다. K는 각 배송지점의 평균값으로 이동하여 각 k에서 가까운 위치의 배송지점으로 그룹화하였고 초기에 정의했던 것처럼 10시에서 20시까지 총 10시간인 10개 구역으로 분할되었다.



<Figure 6> Flowchart for k-Means Clustering



<Figure 7> Grouping Results by Minimum Distance between k and Delivery Points



<Figure 8> Grouping by k-Means Clustering

Moon과 Park[10, 11]의 연구에서 단순히 x, y축을 기준으로 하나의 구역에 동일한 개수로 구역분할 한 것에 비해 k-means clustering 알고리즘을 이용한 구역분할 방법은 근접한 배송지점들이 모여 있는 구역에는 많게, 떨어져있는 배송구역에는 적은 개수의 배송지점이 배정된다. 이것이 보다 합리적인 결과라는 것은 배송구역별로 근접한 지점들은 많은 경우는 많은 개수를, 원거리의 경우는 이동시간의 소요가 길어 단위시간당 배송가능 개수가 적어지기 때문이다.

4.3 인접구역의 결정

차량의 이동경로를 설정하기 위해서는 두 개의 과정이 더 필요하다. 앞과 같이 단위시간 당 배송구역을 설정을 완료했으면, 다음은 Moon과 Park[11]에서와 같이 차량의 이동속도에 변화가 없는 시간구간대로 분할된 구역들을 묶어야 한다. 묶어진 구역 내에서는 원점으로 돌아오지 않는 TSP로 풀어서 최적경로를 구하고, 구해진 지역적 경로를 서로 연결하면 전체 경로가 완성된다. 여기에서 분할된 구역들을 묶은 것을 “인접구역”이라 한다.

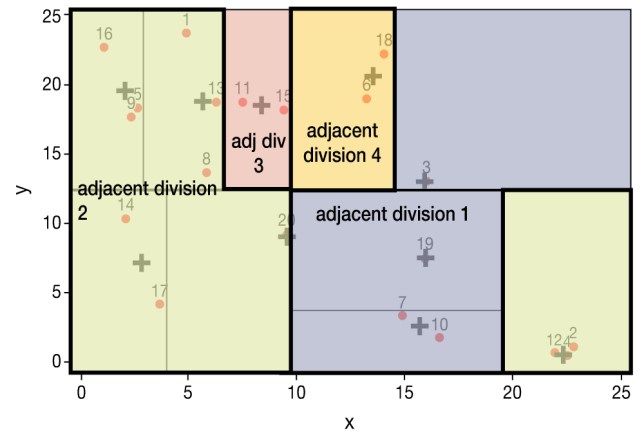
인접구역 결정 알고리즘은 해당 시간대에 배송하면 유리한 구역의 모음을 결정하는 알고리즘이다. 인접구역을 결정할 때에는 구역들 중 현재 배송하는 것이 가장 유리한 구역, 즉 해당 시간대의 속도에 소요시간이 가장 적게 걸리는 구역을 해당시간대 크기에 따라 개수로 맞춘다. 각 시간대의 크기는 Moon과 Park[10, 11]에서와 같이 첫 번째 시간대인 10시부터 12시 59분까지의 3시간인 3개 구역, 두 번째 시간대는 13시부터 17시 59분까지의 5시간인 5개 구역, 세 번째 시간대는 18시부터 18시 59분까지 1시간인 1개 구역, 네 번째 시간대는 19시부터 20시까지의 1시간인 1개 구역으로 나눈 인접구역으로 설정한다.

인접구역의 결정은 다음과 같은 단계에 의거 수행한다. 시간대에 따라 묶어진 배송구역들의 모음을 인접구역 1, 인접구역 2, 인접구역 3, 인접구역 4로 구분하였다.

- [단계 1] 임의로 지정되는 창고에서 각 구역의 k까지의 직선거리를 계산한다.
- [단계 2] 창고에서 각 구역에 속하는 배송지점의 속도평균을 계산한다. 각 시간대에 정해져 있는 속도를 이용해 구역 내의 배송지점에 대한 속도평균을 계산한다. 이는 각 해당하는 시간대에 대한 구역 내의 평균 속도를 계산함으로써 창고 혹은 기준점에서의 이동하는 구역 간 속도를 비교하는 것이다.
- [단계 3] 단계 1과 단계 2의 값을 이용해 시간을 구한다.
- [단계 4] 단계 3에서 구한 시간 중 가장 짧은 시간이 소요되는 구역을 1구역으로 정한다. 총 10개의 구역

중 배송차량이 제일 먼저 방문하게 되는 구역이 된다. <Figure 8>에서의 k10에 여기에 해당된다. 창고에서 k10구역으로 가는 시간이 5.097분으로 가장 짧게 나왔기 때문에 첫 번째 시간대에서는 k10구역을 1번 시간대의 첫 구역으로 정해진다. [단계 5] 선택된 구역의 k를 새로운 시작점으로 정하고 단계 1부터 반복한다. 인접구역 1로 선택된 k10의 k에서 나머지 구역의 k와의 거리와 배송지점의 속도평균을 이용해 시간을 계산하고 그 중 가장 시간이 짧게 소요되는 구역을 다음 구역으로 설정한다. 각 시간대의 개수에 맞춰 인접구역을 할당하고 구역의 속도는 시간대에 맞는 속도를 이용해 평균시간을 계산한다. 인접구역 생성결과 예시는 <Figure 9>와 같다. 그림에서와 같이 인접구역 2의 경우 한 개 구역의 분리현상이 발생하였다. 분리구역을 타인접구역의 것과 교체하면 이러한 현상을 방지할 수는 있으나 차량이동소요시간의 기준으로 평가하면 먼 배송지가 같은 구역에 소속되었다라도 이동소요시간은 오히려 짧을 수 있으므로 추후 연구를 통하여 전체 경로를 평가한 후 판단해야할 문제이다. 총 3시간대인 인접구역 1은 3개의 구역으로 묶어졌고, 두 번째 시간대인 인접구역 2는 5시간이므로 5개의 구역이 배정되었다. 나머지 3번째 시간대인 인접구역 3과 4번째 시간대인 4는 각 한 시간씩이므로 1개씩 할당되었다.

그룹으로 묶인 각각의 인접구역 1, 2, 3, 4 내에서는 차량의 이동속도 변화가 없다고 볼 수 있으므로 일반적인 TSP 해법으로 해를 구할 수 있으며, 이 지역적 경로를 서로 연결하면 전체 이동경로를 만들 수 있게 된다. 지역적 경로와 전체적 차량이동경로를 구성하는 해법은 남겨진 연구 주제이다.



<Figure 9> Final Results by k-Means Clustering

지역적 경로는 각 인접구역 내에서 한 배송점에서 출발하여 모든 배송점을 방문하는 최단경로는 설정하는 것이다. 각 인접구역 내에서는 차량이동속도의 변화가 없는 것으로 가정할 수 있기 때문에 일반적인 TSP 해법을 적용하여 최단경로를 설정할 수 있다. 전체 이동경로는 이들 지역적 경로들을 서로 연결하여 만들면 된다. 추후 연구에서 차량이동경로 설정을 위해 구체적인 지역적 경로 설정 방법과 전체 경로 설정이 용이하게 되도록 하기 위한 방안을 모색할 것이다.

5. 결론

본 논문에서는 이동시간대에 따라 차량의 이동속도가 변동되는 높은 난이도의 VRP 해법개발에 적용할 수 있는 새로운 구역의 분할방법을 설계하고 제안하였다. 제안된 방법은 본문에서 기술한 것과 같이 기존의 연구 대비 보다 합리적인 분할결과를 보여주었다. 기존의 연구에서는 시간당 방문할 수 있는 배송지점의 개수를 동일한 것으로 가정하여 동일한 개수의 배송지점을 구역 당 배정되도록 하였다. 그러나 근접한 배송지점들이 있는 경우 배송지점들 사이의 이동에 소요시간이 짧아져서 동일한 시간에 많은 배송지점을 방문할 수 있으므로 해당 구역은 보다 많은 개수가 배정되도록 해야 하는 것이 보다 개선된 해의 도출에 도움이 되기 때문이다.

Acknowledgement

이 논문은 동아대학교 교내연구비 지원에 의하여 연구되었음.

References

- [1] Donati, A.V., Montemanni, R., Casagrande, N., Rizzoli, A.E., and Gambardella, L.M., Time dependent vehicle routing problem with a multi ant colony system. *European Journal of Operation Research*, 2006, Vol. 185, No. 3, pp. 1174-1191.
- [2] Haghani, A. and Jung, S., A dynamic vehicle routing problem with time-dependent travel times. *Computers and Operations research*, 2005, Vol. 32, No. 11, pp. 2959-2986.
- [3] Jabali, O., Van Woensel, T., de Kok, A.G., Lecluyse, C., and Peremans, G., Time-dependent vehicle routing subject to time delay perturbations. *IIE Transactions*, 2009, Vol. 42, pp. 1049-1066.
- [4] Kenyon, A.S. and Morton, D.P., Stochastic vehicle routing with random travel times. *Transportation Science*, 2003, Vol. 37, No. 1, pp. 69-82.
- [5] Kuo, Y., Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers and Industrial Engineering*, 2010, Vol. 59, No. 1, pp. 157-165.
- [6] Kuo, Y., Wang, C.C., and Chuang, P.Y., Optimizing goods assignment and the vehicle routing problem with time-dependent travel speeds. *Computer and Industrial Engineering*, 2008, Vol. 57, pp. 1385-1392.
- [7] Lloyd, S.P., Least square quantization in PCM. *IEEE Transactions on Information Theory*, 1957, Vol. 28, No. 2, pp. 129-137.
- [8] MacQueen, J.B., Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281-297.
- [9] Malandraki, C. and Daskin, M.S., Time Dependent Vehicle Routing Problems : Formulations, Properties and Heuristic Algorithm. *Transportation Science*, 1992, Vol. 26, No. 3, pp. 185-200.
- [10] Moon, G. and Park, S., Analysis and reconstruction of vehicle speeds to design an efficient time dependent VRP heuristic. *Journal of the Society of Korea Industrial and Systems Engineering*, 2012, Vol. 35, No. 1, pp. 140-147.
- [11] Moon, G. and Park, S., A Possible heuristic for variable speed vehicle routing problem with 4 time zone. *Journal of the Society of Korea Industrial and Systems Engineering*, 2012, Vol. 35, No. 4, pp. 171-178.

ORCID

Je-Min Ha | <http://orcid.org/0000-0002-4787-9628>
 Geeju Moon | <http://orcid.org/0000-0003-1546-6316>