

배달과 수집을 수행하는 차량경로문제 휴리스틱에 관한 연구 : 수도권 레미콘 운송사례

지창훈* · 김미이* · †이영훈*

Heuristic for the Pick-up and Delivery Vehicle Routing
Problem : Case Study for the Remicon Truck Routing in the
Metropolitan Area

Chang Hun Ji* · Mi-Yi Kim* · †Young Hoon Lee*

■ Abstract ■

VRP (Vehicle Routing Problem) is studied in this paper, where two different kinds of missions are to be completed. The objective is to minimize the total vehicle operating distance. A mixed integer programming formulation and a heuristic algorithm for a practical use are suggested. A heuristic algorithm consists of three phases such as clustering, constructing routes, and adjustment. In the first phase, customers are clustered so that the supply nodes are grouped with demand nodes to be served by the same vehicle. Vehicle routes are generated within the cluster in the second phase. Clusters and routes are adjusted in the third phase using the UF (unfitness) rule designed to determine the customers and the routes to be moved properly. It is shown that the suggested heuristic algorithm yields good performances within a relatively short computational time through computational experiment.

Keywords : Vehicle Routing Problem, Clustering, Constructing Routes

논문접수일 : 2006년 11월 24일 논문게재확정일 : 2007년 08월 14일

* 연세대학교 정보산업공학과

† 교신저자

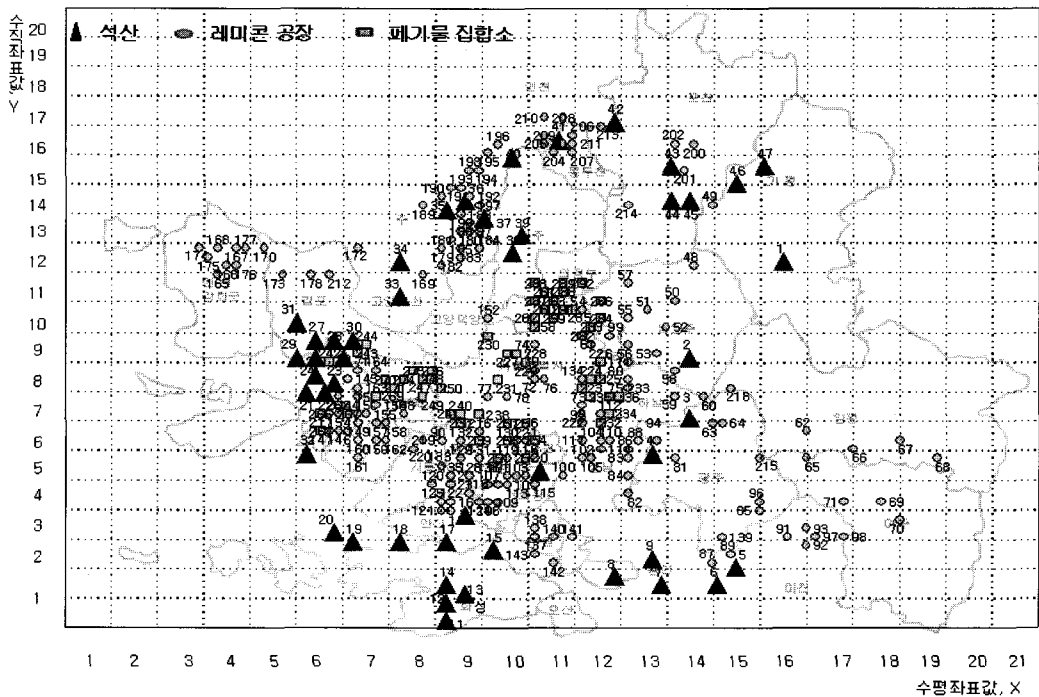
1. 서론

일반적인 차량경로문제(Vehicle Routing Problem)는 차고지(depot)에서 출발한 차량이 배달이나 수집 등의 서비스를 요구하는 고객들을 모두 방문하고 차고지로 돌아오는 데 소요되는 이동거리를 최소화하는 차량경로를 결정하는 문제이다. 차량은 통상적으로 차고지를 출발하여 차량의 용량이 허용될 때까지 고객들을 서비스하고 차고지로 귀환하며 배달이나 수집 중 하나의 서비스만을 제공한다.

본 연구에서는 일반적인 차량경로문제와는 달리, 차량이 일과시간동안 한 번 이상 사용가능(multiple use of vehicles)하며 다수의 차고지에서 출발한 차량이 배달을 요구하는 고객들을 서비스하고 차고지로 복귀하는 길에 수집을 요구하는 고객들을 서비스하는 두 가지의 임무를 수행하는 문제를 다루고자 한다. 이러한 문제는 전자재 운송업과 같이 차고지를 출발하는 차량의 수송자재와 차고지

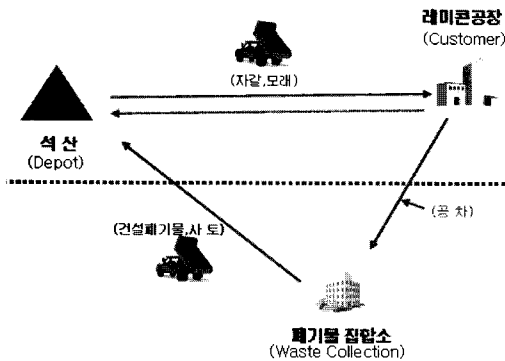
로 귀환할 차량의 수송자재가 상이하며, 한 차량에 적재된 자재들을 한 고객에게 모두 배달하는 경우에 적용할 수 있다. 즉, 일반적인 차량경로문제에서는 모든 고객들이 하나의 차량에 의해 한번 서비스를 받을 수 있었으나 해당 문제는 모든 고객들이 여러 번 서비스 받을 수 있다는 것을 의미한다.

<그림 1>은 수도권에서 운영되고 있는 레미콘 전자재 운송업 현황으로서 현재 수도권에는 대략 석산 40개, 레미콘 공장 120개와 폐기물 집합소 40개가 산재하고 있다. 전자재 물류업체들은 석산의 모래, 자갈을 레미콘 공장까지 실어 나르기 위해 덤프트럭을 이용하는데 매일 일정 대수의 덤프트럭이 석산과 레미콘 공장 사이를 왕복하며, 석산의 골재채취 부분을 메우기 위해 또 다른 덤프트럭이 폐기물 집합소와 석산사이를 왕복한다. 여기서 석산은 차량경로문제에서의 차고지, 레미콘 공장은 배달을 요구하는 고객, 폐기물 집합소는 수집을 요구하는 고객을 의미한다. <그림 1>에서와 같이 석



<그림 1> 수도권의 석산, 레미콘 공장, 폐기물 집합소의 배치도

산은 지형적인 특성 때문에 수도권 외곽에 불균형적으로 분포하고 있으며, 석산으로부터 건자재를 공급받고 있는 레미콘 공장은 수도권 전역에 고루 분포되어 있다. 즉, 석산에서 출발한 차량이 어느 레미콘 공장에 건자재를 공급하느냐와 석산으로 귀환하기 전에 어느 폐기물 집합소를 거치느냐에 따라 차량이동거리가 상이해지므로 적절하게 석산, 레미콘 공장, 폐기물 집합소가 하나의 그룹으로 구분되어 차량경로를 생성해야 한다.



<그림 2> 덤프트럭의 운송경로

본 연구는 40여개의 권역으로 나누어 서비스하고 있는 실제 수도권 레미콘 운송사례를 토대로 진행되며, <그림 2>와 같이 두 가지의 임무를 동시에 고려하여 통합 배차함으로써 석산에서 레미콘 공장까지 이동한 차량이 공차로 복귀하는 것이 아니라 근처에 있는 폐기물 집합소를 경유하여 폐기물 집합소의 수집요구량을 충족시키고 복귀하도록 하여 차량의 이동거리를 단축시키고자 한다.

<그림 1>과 같이 여러 개의 석산(Multi-Depot)이 존재하며, 석산에는 25톤의 동일한 덤프트럭(homogeneous)이 존재하며, 석산별로 운용 가능한 덤프트럭의 수가 정해져있다. 그리고 덤프트럭은 일과시간내에서 여러 번 운행가능(Multi-Trip)하고, 레미콘 공장으로 이동한 덤프트럭이 폐기물집합소의 폐기물을 석산까지 이동시켜야하므로 석산은 차고지가 될 뿐 아니라 또 다른 수요지가 된다는 점에서 기존 문제와 차이가 있다. 또한 덤프트럭은

자갈이나 모래를 레미콘 공장에 나누어 내릴 수 없고 덤프트럭의 용량만큼 내리거나 실어야 한다. 즉, 한 대의 덤프트럭은 석산을 출발하면 하나의 레미콘 공장과 폐기물 집합소를 서비스할 수 있다. 그리고 폐기물 집합소의 수요량이 레미콘 공장의 수요량보다 작기 때문에 일부 덤프트럭은 폐기물 집합소를 거치지 않고 레미콘 공장에서 석산으로 바로 복귀하게 된다.

차량경로문제에 대한 전통적인 모형은 Danzig and Ramser[7]가 최초로 제안했으며, 차량경로문제에서 파생되어 나온 수집과 배달에 대한 요구를 가진 차량경로문제는 Pisinger and Ropke[11]가 언급하였듯이 대략 다음 세 가지로 나누어 볼 수 있다. 첫 번째는 VRPB(Vehicle Routing Problem with Backhauls)로 먼저 배달에 대한 요구를 충족시킨 후 돌아오는 길에 수집을 하는 문제로 Brandao[6]는 타부탐색을 이용하여 해를 제시하였으며, Toth and Vigo[13]는 클러스터링 후 TSP(Traveling Salesman Problem)해법으로 초기해를 생성하고, 해를 개선하는 방법을 제안하였다. 두 번째로 VRPMDP (Vehicle Routing Problem with Mixed Delivery and Pick-up) 문제로 배달과 수집을 원하는 고객이 따로 존재하여 각각을 서비스를 하는 차량경로 문제이다. 최근에는 Wade and Salhi[14]가 back-haul 노드를 haul 노드에 삽입시키는 휴리스틱을 제시하여 효과적으로 이 문제를 해결하였다. 세 번째 모델로는 수집과 배달을 동시에 수행하는 VRP SDP(Vehicle Routing Problem with Simultaneous Delivery and Pick-up)문제로 Bianchessi and Righini[5]가 첫 단계에서 초기해를 만들고, 두 번째 단계에서 타부탐색을 기반으로 arc-exchange와 node-exchange를 이용하여 이웃해를 형성하고 개선하는 휴리스틱을 제시하여 이 문제를 해결하였다. 정은용 외[4]는 VRPMDP 문제에 대해 성능의 개선이 이루어진 유전 알고리즘을 제시하였으나 해의 계산시간에서는 개선을 이루지 못하였다. 장희정 외[3]는 시간제약과 하역장 용량제약이 있는 차량경로문제에 대해 자주 사용되는 메타휴리스

스틱으로서 타부 탐색을 사용하였다. 또한 Nagy and Salhi[9]는 VRPMDP, VRPSDP의 해를 제공할 수 있는 insertion 기반의 휴리스틱을 제시하였다.

그리고 앞에 언급된 연구들은 하나의 차고지를 지닌 상황을 가정하였으나 Oukil et al.[10], Salhi and Sari[12] and Irmich[8]는 여러 개의 차고지를 지닌 차량경로문제에 대한 해법을 제시하였다. Oukil et al.[10]는 column generation 방법을 통해 해를 제시하였고, Salhi and Sari[12]는 상이한 차량을 지닌 상황을 가정하였다. Irmich[8]는 배달과 수집을 수행하는 형태로 문제를 정의하고 이에 대한 해를 제시하였다. 비슷한 형태의 차량경로문제에 대해 최적해를 도출하기 위한 알고리즘을 제시한 강자영 외[1]는 배달과 수집이 있고 시간제약이 있는 모델에서 Branch and Price 알고리즘과 열거법 등을 사용하였다. 이원동 외[2]는 본 논문에서 다루고 있는 문제와 유사한 모델에 대한 연구로서 동일한 기초데이터를 사용하여 협업적 차량운송관리에 대한 내용을 분석하고 있다. 이는 현재 40여개가 넘는 사업자에 의해 운영되고 있는 전자재 물류시장의 효율화를 위해 10개 이하의 컨소시엄을 구성하는 것이 효율적이라는 전제아래 여러 대안에 대한 분석을 시도하였다. 따라서 전략적인 의사결정을 위한 분석으로 최적화 기법을 근간으로 한 해법을 적용하였다.

본 논문은 적절한 형태로 협업이 이루어지고 있다는 전제아래 실제적으로 차량운송이 이루어지는 운영적 해법에 대해 제시하고자 한다. 이는 매일 이루어지는 차량운송 스케줄의 생성에는 최적화 기법보다는 짧은 시간 내에 해법이 도출되고 담당자가 수시로 변경, 유지보수가 가능한 수준의 실제적 해법을 제시하는데 목적을 두고 있다. 다만 필요에 따라서는 작은 그룹의 고객에 대한 최적화 운영계획 알고리즘을 제시하고 발견적기법과 비교하여 성능을 분석함으로써 발견적기법의 우수성을 검증하고자 한다. 본 논문은 기존의 수집과 배달 문제와 여러 개의 차고지를 지닌 차량경로문제와 몇 가지 측면에서 다르다. 첫째, 수집과 배달을 원하

는 고객들이 균형적으로 분포되어 있는 상황이 아니며 둘째, 출발지에서 운행을 시작한 차량이 여러 번 출발이 가능한 Multi-Trip 문제이고 셋째, 서비스를 제공할 석산들의 집합이 정해진 경우가 아니라 적절한 형태로 구분하여 이에 대한 운송경로를 구하는 두 가지 문제를 동시에 해결하고자 한다. 우선 위 문제에 대해 수리적 모형을 제시하고 실행 가능한 시간 내에 최적해를 구할 수 없기 때문에 이에 대한 발견적 기법을 제시하여 다양한 해를 구하고 이에 대한 비교 검토를 하고자 한다.

2. 수리 모형

2.1 기호

i	: 석산($i = 1, \dots, I$)
j	: 레미콘 공장($j = 1, \dots, J$)
k	: 폐기물 집합소($k = 1, \dots, K$)
N_i	: 석산 i 의 차량의 수
l	: 차량($l = 1, \dots, N_i$)
m	: 트립($m = 1, \dots, M$)
S_i	: 석산 i 에서의 공급 가용차량 대수
D_j	: 레미콘 공장 j 에서의 요구량을 배달하는데 필요로 하는 차량의 대수
M_k	: 폐기물 집합소 k 에서의 이동요구량을 수집하는데 필요로 하는 차량의 대수
d_{ij}	: 석산 i 에서 레미콘 공장 j 로의 이동거리
d_{jk}	: 레미콘 공장 j 에서 폐기물 집합소 k 로의 이동거리
d_{ki}	: 폐기물 집합소 k 에서 석산 i 로의 이동거리
d_{ji}	: 레미콘 공장 j 에서 석산 i 로의 이동거리
X_{ijm}	: m 번째 트립에 석산 i 에서 출발한 차량 l 이 레미콘 공장 j 로 이동하면 1, 아니면 0
Y_{ijkm}	: m 번째 트립에 석산 i 에서 출발한 차량 l 이 레미콘 공장 j 로 이동하고 폐기물 집합소 k 로 이동하면 1, 아니면 0

본 문제유형의 특성은 전체 차량의 총이동거리

를 최소화 시키는 것이므로 모든 차량의 이동거리를 목적함수에서 계산한다. 석산에서 출발한 차량은 레미콘 공장으로 이동 후 석산으로 복귀하거나, 레미콘 공장에서 폐기물 집합소로 이동하는 두 가지 경우를 생각할 수 있다. 물론 폐기물 집합소로 이동한 차량은 모두 석산으로 다시 복귀한다. 따라서 목적함수는 석산에서 레미콘 공장까지 이동한 거리에 레미콘 공장에서 폐기물 집합소를 거쳐 다시 석산으로 복귀하는 차량의 이동거리를 더한다. 마지막으로 레미콘 공장에서 석산으로 복귀한 차량의 이동거리를 더하면 된다.

제약식에서는 석산에서 나가는 차량은 레미콘 공장의 배달 요구량과 폐기물 집합소의 수집 요구량을 모두 만족시켜 주어야 하며, 차량의 대수는 레미콘 공장에서 들어오는 차량과 폐기물 집합소를 거쳐서 들어오는 차량의 합과 같아야 한다. 또한 레미콘 공장에 들어온 차량은 석산으로 복귀하거나, 폐기물 집합소로 이동하여야 한다. 즉 모든 요구량을 만족시키면서 차량의 입출량 대수도 일치하여야 한다. 또한 각 석산에서의 차량 가용대수, 차량 당 하루 운행회수(Multi-trip의 수), 석산 공급가용량이 정해져 있어 이들을 초과할 수 없는 특징을 지니고 있다. 이를 수식화하면 다음과 같다.

2.2 최적화 수리모델

$$\text{Minimize } \sum_{i,l,j,m} X_{iljm} \times d_{ij} + \sum_{i,l,j,k,m} Y_{iljkm} \times (d_{jk} + d_{ki}) + \sum_{i,l,j,m} X_{jljm} \times d_{ji} \quad (1)$$

$$\sum_l \sum_j \sum_m X_{iljm} = \sum_l \sum_j \sum_m X_{jljm} + \sum_l \sum_j \sum_k \sum_m Y_{iljkm} \quad \forall i \quad (2)$$

$$\sum_i \sum_l \sum_m X_{iljm} = \sum_i \sum_l \sum_m X_{jljm} + \sum_i \sum_l \sum_k \sum_m Y_{iljkm} \quad \forall j \quad (3)$$

$$\sum_i \sum_l \sum_m X_{iljm} = D_j \quad \forall j \quad (4)$$

$$\sum_i \sum_l \sum_j \sum_m Y_{iljkm} = M_k \quad \forall k \quad (5)$$

$$\sum_l \sum_j \sum_m X_{iljm} \leq S_i \quad \forall i \quad (6)$$

$$\sum_l \sum_j X_{iljl} \leq N_i \quad \forall i \quad (7)$$

$$X_{iljm}, Y_{iljkm} = 0 \text{ or } 1 \quad \forall i,l,j,k,m \quad (8)$$

최적화 수리모델의 식 (1)은 목적함수로 각 차량의 총 이동거리, 즉 석산으로부터 레미콘 공장으로의 이동, 레미콘 공장에서 폐기물 집합소를 거쳐 석산으로 귀환하는 경로 및 폐기물 집합소를 거치지 않고 바로 귀환하는 경로에 대한 총 운행거리를 최소화 한다. 식 (2)는 석산에서 출발한 차량은 레미콘 공장에서 공차로 복귀하는 차량과 폐기물 집합소에서 수집량을 적재하고 복귀하는 차량의 합과 같음을 나타낸다. 식 (3)은 모든 레미콘 공장에 대하여 들어온 차량은 석산으로 복귀하거나 폐기물 집합소로 나가게 됨을 나타낸 식이다. 식 (4)는 모든 레미콘 공장에 대한 배달요구량 충족 조건이며, 식 (5)는 모든 폐기물 집합소에 대한 수집요구량 충족조건을 나타낸 식이다. 식 (6)은 석산에서 공급받는 차량의 총 대수는 석산의 공급 가용차량의 수와 같거나 작아야 한다는 의미이며, 식 (7)은 석산의 차량대수에 관한 제약식이고 식 (8)은 결정변수가 0 또는 1이라는 제약식이다.

3. 발견적 알고리즘

차량경로문제가 NP-hard 조합최적화문제로 노드의 수가 늘어남에 따라 최적해를 구하기 위한 계산시간이 급격히 증가한다. 따라서 작은 사이즈의 문제에 대해서는 위에서 제시한 최적화 수리모델을 이용하면 쉽게 최적해를 구할 수 있지만, 큰 사이즈의 실제 문제에서는 최적해를 구하기 어렵다. 일반적으로 큰 사이즈의 문제는 여러 개의 작은 사이즈의 클러스터로 나누어 해를 구할 수 있는데 클러스터를 나누는 방법에 따라 해의 성능이 달라진다.

차량경로문제는 대체적으로 우선 클러스터로 나누고 나누어진 클러스터 내의 고객에 대하여 경로

를 구하는 대표적인 알고리즘으로는 Saving, Sweeping, Petal method 등이 있다. 이러한 방법들은 하나의 차고지(원점)를 중심으로 클러스터를 생성하기 때문에 여러 개의 석산(차고지)을 지닌 문제의 경우에는 실행 불가능한 클러스터를 생성할 수 있다. 따라서 본 논문에서는 무계중심이라는 개념을 제안하여 석산들의 생산량을 기준으로 가상의 원점을 계산하고 이에 맞게 고객들의 좌표를 수정하였다. 그리고 Sweeping 방법을 적용하여 석산들만을 포함하는 클러스터를 생성하고 클러스터에 적합한 고객(레미콘 공장과 폐기물 집합소)들을 삽입하였다.

여러 개의 클러스터가 형성된 후 제 2장에서 제시한 수리모형을 이용하여 최적해를 구할 수 있으나 하나의 클러스터내의 고객의 수가 많이 존재하는 실제 상황에서는 최적해를 구하는데 많은 시간이 소요되거나 최적해를 구할 수 없는 경우가 존재한다. 따라서 짧은 시간 내에 차량의 경로를 생성할 수 있도록 NAS(Nearest Available Search) 방법을 제시하였다. 또한 이를 통해 얻어진 초기해는 Adjustment 방법을 이용하여 클러스터별로 부적합 고객들을 선택하여 이웃클러스터로 삽입하여 해를 개선한다. 부적합 고객들을 선택하기 위해 UF(Unfitness) 규칙을 사용하였다.

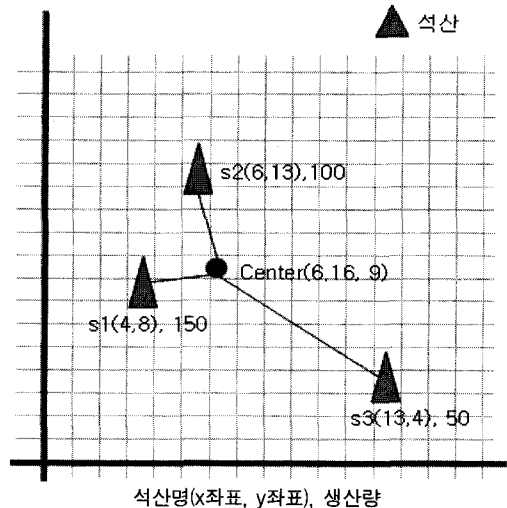
제시한 발견적 해법은 CRA(Clustering-Route constructing-Adjustment)라 명명하였으며, 본 논문에서 제시한 클러스터링방법과 제 2장에서 제시한 수리모형을 토대로 얻어진 차량경로해의 조합으로 구성된 방법을 FCSO(First Clustering Second Optimization)라 명명하여 CRA를 통해 얻어진 해의 결과와 비교하고자 한다.

CRA 방법과 FCS 방법 모두에서 사용되는 클러스터 분할 방법에서 사용된 무계중심은 클러스터 내에서 각 석산들의 실제 공급가용량이 모두 모여 있다고 생각하는 가상의 점이다. 클러스터 내에 속해 있는 석산들의 위치를 $(x_i, y_i), i = 1, 2, \dots, p$ 라고 하고 석산의 공급가용량을 $C_i(i = 1, 2, \dots, p)$ 라고 하면 무계중심 (x, y) 는 다음과 같이 구할 수 있다.

$$(x, y) = \left(\frac{\sum_{i=1}^p C_i x_i}{\sum_{i=1}^p C_i}, \frac{\sum_{i=1}^p C_i y_i}{\sum_{i=1}^p C_i} \right)$$

무계중심은 석산의 실제 공급가용량에 비례해서 계산되므로 공급량이 큰 석산 가까이에 위치하게 된다. 무계중심의 개념을 이용하게 되면 클러스터에 포함시킬 석산을 선택하는데 있어 기존의 무작위적인 거리나 각도를 이용한 Sweeping 방법에 비해 공급을 많이 해줄 수 있는 석산에 적정량만큼 가까운 고객을 할당시키므로 초기해를 구성할 때 최적해에 가깝도록 선택할 수 있고, 최적해에 근사한 초기해를 구하여 전체적인 계산시간을 절감할 수 있다. 예를 들면 <그림 3>과 같이 3개의 석산이 있고, 각각의 석산 s1, s2, s3의 공급량이 150, 100, 50이라고 하면 무계중심(x, y)는 (6.16, 9)가 된다. 즉 무계중심은 실제 공급 가용량이 큰 석산 쪽으로 옮겨가게 되는 것이다. 이것은 석산의 개수가 여러 개일 때도 동일하게 적용된다.

$$(x, y) = \left(\frac{150}{300} \times 4 + \frac{100}{300} \times 6 + \frac{50}{300} \times 13, \frac{150}{300} \times 8 + \frac{100}{300} \times 13 + \frac{50}{300} \times 4 \right) = (6.16, 9)$$



<그림 3> 무계중심 계산 예제

제시한 CRA 방법의 3단계는 해의 개선과정으로

각 클러스터의 고객이 현재의 클러스터에 속해 있는 것이 얼마나 적절한지를 수치화하여 개선의 여지가 큰 고객을 선정하고 이를 재조정하는 방법으로 진행한다. 해의 개선 시 형성된 클러스터에서 하나의 고객을 뽑아 이웃 클러스터로 보내게 되는데, 현재의 고객이 해당 클러스터 내에서의 부적합도를 정의한 UF(Unfitness) 규칙을 이용한다. UF 지수는 클러스터의 무게중심으로부터 고객이 멀리 떨어져 있으면 있을수록, 각도차가 크면 클수록 이웃 클러스터로 이동시켜야 되는 고객들로 볼 수 있고, 원점과의 거리가 가까울수록 반대쪽에 있는 클러스터로 이동시킬 때 더 좋은 해를 구할 수 있음을 반영한다. 가상의 원점을 기준으로 클러스터들이 형성되었기 때문에 '원점과의 거리가 짧다'라는 의미는 '반대쪽에 형성된 클러스터와의 거리가 짧다'라는 의미로 볼 수 있다. 따라서 UF 지수는 거리 차, 각도 차, 원점과의 거리 차의 세 가지 요소로 구성하였으며 모든 고객들에 대해 UF 지수를

계산한 후에 지수가 가장 큰 고객을 부적합 고객으로 선택하여 이웃 클러스터로 이동시키는 과정을 반복한다.

UF 지수는 무게중심과의 거리 차, 각도 차, 원점과의 거리 세 가지 요소를 고려하였다. 하지만 세 가지의 단위를 맞추주기 위해 각도 차를 호의 길이로 변환하여 사용하였다. 모든 레미콘 공장과 폐기물 집합소에 대하여 무게중심과의 거리 차(Δf), 호의 길이($\Delta \alpha$), 원점과의 거리(Δd)를 계산하여 다음과 같이 나타낼 수 있다. 그리고 α , β , σ 는 이 세 가지 요소들의 서로 다른 가중치를 나타내는 값으로 이 값들에 따라 고객들의 UF 지수가 변경되어 다양한 이웃해를 생성하고 탐색하게 된다.

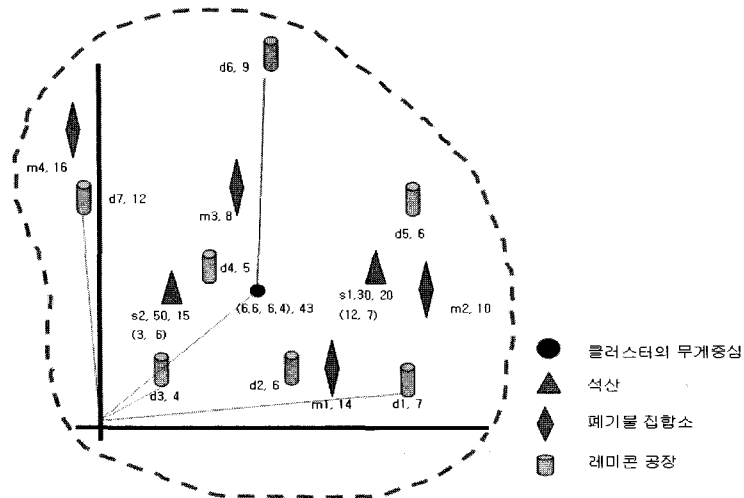
예를 들면 <그림 4>처럼 7개의 레미콘 공장들이 하나의 클러스터에 포함되었다고 하자. 이 클러스터에서 인접 클러스터로 이동시켜야 할 레미콘 공장을 뽑는다면 무게중심에서 가장 거리가 먼 d_6 , 무게중심과 원점이 이루는 선분과 각도 상으로 가

$$UF = \alpha \times \Delta f + \beta \times \Delta \alpha - \sigma \times \Delta d \quad (\text{단, } \alpha + \beta + \sigma = 1)$$

$$\Delta f = \sqrt{(\text{무게중심의 } x \text{ 좌표} - \text{레미콘공장의 } x \text{ 좌표})^2 + (\text{무게중심의 } y \text{ 좌표} - \text{레미콘공장의 } y \text{ 좌표})^2}$$

$$\Delta \alpha = |\angle \text{무게중심} - \angle \text{레미콘공장} \times R \quad (\text{단, } R \text{은 반지름으로 일정한수})$$

$$\Delta d = \sqrt{(\text{레미콘공장의 } x \text{ 좌표})^2 + (\text{레미콘공장의 } y \text{ 좌표})^2}$$

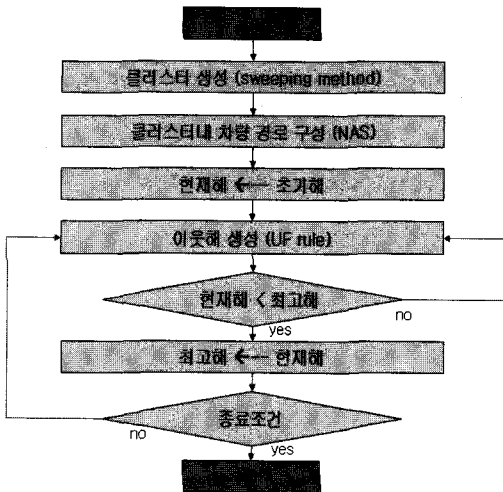


<그림 4> UF 지수의 개념

장 멀리 떨어진 d7이나 d1, 원점과의 거리가 가장 가까운 d3 중에서 하나가 될 것이다. 여기서 클러스터는 원점을 기준으로 형성되어 있기 때문에 원점과의 거리가 짧다는 것은 원점의 반대편에 있는 클러스터와의 거리가 상대적으로 다른 레미콘 공장들보다 짧다는 의미로 해석할 수 있다. UF 지수는 이 세 가지 요소를 고려하면서 레미콘 공장을 선택하되 다양한 레미콘 공장이 번갈아가면서 선택되도록 각각의 요소에 가중치를 두고, 실험 시 가중치를 변경시킴으로써 다양한 이웃해로 이동할 수 있도록 설계했다.

3.1 CRA 알고리즘

<그림 5>는 CRA 알고리즘의 진행과정을 표시한 것이다.



<그림 5> CRA 알고리즘 flow-chart

3.1.1 Clustering(1 단계)

• Step 1 : 클러스터에 포함될 석산 선택

석산들의 좌표 및 생산량을 기준으로 전체의 무게중심을 구한다. 이 무게중심을 가상의 원점으로 하여 모든 노드(석산, 레미콘 공장, 폐기물 집합소)들의 좌표를 수정한다. 그리고 가상의 원점과 노드들의 이루는 각도를 계산하여 오름차순으로 정렬

한다. 클러스터에 포함될 석산수를 아래의 수식을 이용하여 결정하고, 각도가 제일 작은 석산부터 결정된 수만큼 클러스터에 추가한다. 그리고 각 클러스터에 삽입된 석산들의 좌표와 생산량을 기준으로 각 클러스터의 무게중심을 계산한다.

$$\begin{aligned} & \text{클러스터에 포함될 석산의 수} \\ & = \text{총 석산의 수} / \text{클러스터의 수} \end{aligned}$$

• Step 2 : 클러스터별 실제 공급가용량 계산

각 클러스터내의 석산에서의 공급가용량과 차량 대수를 고려하여 실제 공급해 줄 수 있는 양(RSQ : Real Supply Quantity)을 계산한다. 각 석산에서의 공급가용량과 석산에 소속된 차량이 실어 나를 수 있는 양을 비교하여 작은 양을 실제 공급가용량으로 한다.

$$RSQ = \min(\text{석산의 공급가용량}, \text{차량의 적재용량} \times \text{석산에 포함된 차량수})$$

• Step 3 : 클러스터별 적정배달량과 수집량 결정

Step1에서 계산된 각 클러스터의 무게중심은 레미콘 공장과 폐기물 집합소를 클러스터에 할당할 때의 기준이 된다. 각 클러스터에 어느 정도의 레미콘 공장과 폐기물 집합소를 포함시킬지를 결정하기 위하여 클러스터의 실제 공급량을 기준으로 레미콘 공장(AD : Adequate Demand)의 적당량과 폐기물 집합소(AM : Adequate Move)의 적당량을 구하게 되는데 이는 레미콘 공장과 폐기물 집합소가 일부의 클러스터에 집중되지 않고, 전체적으로 균등하게 할당하기 위함이다. 적당량은 전체 석산의 실제 공급량에 대한 클러스터내의 실제공급량의 비율과 전체 레미콘 공장의 요구량과 클러스터내에서의 요구량을 비교하여 구한다. 그리고 모든 차량은 레미콘 공장을 거친 후에 폐기물 집합소로 이동하므로 폐기물 집합소의 적당량은 레미콘 공장의 적당량을 넘을 수 없다.

$$AD = \frac{RSQ}{\sum_i S_i} \times J, \quad J = \text{총 레미콘공장의 수}$$

$$AM = \min\left(\frac{RSQ}{\sum_i S_i} \times K, AD\right),$$

K = 총 폐기물 집합소의 수

- Step 4 : 클러스터에 포함될 고객 선택
(레미콘 공장과 폐기물 집합소)

각 클러스터의 무게중심으로부터 거리가 가까운 순서대로 레미콘 공장과 폐기물 집합소를 클러스터에 포함시킨다. 이때 용량의 기준은 위에서 구한 AD 와 AM 을 고려하여 이를 초과하기 전까지 계속해서 고객들을 포함시킨다.

- Step 5 : 다음 클러스터 생성

클러스터가 완성되었으면 다음 클러스터를 형성하는데 다음 클러스터의 무게중심보다 각도가 작은 레미콘 공장이나 폐기물 집합소가 있으면 우선적으로 새로 생성되는 클러스터에 추가한다. 이러한 방법으로 모든 석산과 레미콘 공장, 폐기물 집합소가 모두 포함될 때까지 반복하여 클러스터를 완성한다.

3.1.2 Route constructing(2단계)

Clustering(1단계)을 통해 얻어진 클러스터를 토대로 클러스터별로 차량경로를 결정한다. 클러스터 내의 경로를 결정하기 위해서 NAS(nearest available search)방법을 사용하는데 이는 거리를 기준으로 가장 가까이 있는 노드끼리 연결하여 묶어주는 가장 단순한 방법으로 계산시간이 짧으면서도 효과적인 방법이다.

석산 중 실제 공급가능량이 가장 큰 석산을 선택한다. 그리고 다음의 과정을 수행한다.

- Step 1 : 석산, 차량과 레미콘 공장의 선택

석산의 첫 번째 차량을 선택한 후, 가장 가까운 레미콘 공장을 선택하여 배달량이 있으면 차량을 보내고 배달량이 없으면 다른 레미콘 공장을 찾는다. 레미콘 공장으로 차량이 이동한 후에는 배달량에서 차량의 적재용량만큼을 감하여주고, 차량 정

보를 갱신해준다.

- Step 2 : 폐기물 집합소의 선택

가장 가까운 폐기물 집합소를 찾아서 수집량이 있는지를 확인하고, 있으면 차량을 이동시킨 후 차량의 적재용량만큼 수집량을 감한다. 그리고 차량을 석산으로 복귀시킨다. 만약 수집량이 있는 폐기물 집합소가 없으면 차량은 차고지로 복귀한다.

- Step 3 : 석산변경 및 차량의 변경

같은 석산의 두 번째 차량도 이와 같이 운행하며, 선택된 석산의 모든 차량을 사용했을 경우에는 다음으로 실제 공급가능량이 큰 석산을 선택하여 Step 1과 Step 2를 반복 수행한다.

- Step 4 : 차량들의 가용 운행횟수(Trip) 갱신

모든 석산의 차량을 1번씩 이용하였으나 고객들에게 충족시키지 못한 배달량이나 수집량이 남아 있는 경우, 석산들의 모든 차량의 운행 수(trip)를 갱신한다. 그리고 모든 고객들의 충족시키지 못한 배달량이나 수집량이 0이 될 때까지 Step 1, 2, 3, 4를 반복한다.

석산의 차량을 한 번씩만 사용한 후 다른 석산의 차량을 사용하는 이유는 최대한 수요지에서 가까운 석산의 차량을 사용할 가능성을 높이기 위함이다. 이러한 과정을 거쳐 얻어진 해가 초기해가 된다.

3.1.3 Adjustment(3단계)

초기해를 더 좋은 해로 개선시키기 위해서는 클러스터 내에서 가장 부적합한 고객을 뽑아서 인접 클러스터로 이동시켜야 하는데 이를 선택하는 기준이 UF 규칙으로 α, β, σ 값에 따라 다양하게 부적합 고객을 선택한다.

- Step 1 : α, β, σ 값 변경 및 UF 지수 계산

가중치인 α, β, σ 값을 변경하여 구성된 모든 클

러스터 내에서 모든 레미콘 공장과 폐기물 집합소에 대하여 UF 지수를 계산한다.

• Step 2 : 이웃해의 값 계산

UF 규칙을 적용하여 계산한 레미콘 공장과 폐기물 집합소들을 클러스터 내에서 각각 내림차순으로 정렬하고, 각 클러스터에서 첫 번째 레미콘 공장들을 선정하여 좌측의 클러스터로 이동시켜 목적값을 계산하고, 우측의 클러스터로 이동시켜 목적값을 계산하여 현재해와 비교한다. 좌우측으로 이동시키는 이유는 선택된 레미콘 공장이 바로 옆에 있는 클러스터로 이동하였을 때 해의 개선 가능성이 높기 때문이다. 이와 같은 방법으로 폐기물 집합소에 대해서도 이웃해를 탐색한다.

• Step 3 : 최선해 개선 및 종료조건

현재해가 최선해보다 좋으면 현재 해를 최선해로 개선하고, 그렇지 않으면 종료조건이 될 때까지 반복 실행한다. 종료조건은 일정반복 횟수동안 해의 개선이 없거나 자체반복 횟수가 일정수를 넘는 경우이다.

3.2 FCSO 알고리즘

2단계의 알고리즘으로 구성되며, 첫 번째 단계는 무게중심을 이용하여 여러 개의 클러스터를 구성하는 과정으로 위에서 설명한 CRA 알고리즘과 동일하고, 두 번째 단계에서의 최적화하는 방법은 결정된 클러스터에 대하여 위에서 제시한 최적화 모델 수식을 ILOG사의 OPL 상용프로그램으로 계산하여 해를 구한다.

이 알고리즘은 위에서 설명한 CRA 알고리즘의 클러스터를 생성하는 방법과 동일하지만 클러스터 내에서 최적경로를 탐색하는 방법이 다르다. OPL에서는 모든 경우의 수를 고려하여 계산하기 때문에 시간은 오래 걸리지만 더 좋은 해를 구할 수 있다. 특히 문제에서 노드의 숫자가 60개 이상 넘어 가면서부터 급격히 증가하여 노드가 100개를 넘어

가면 해를 찾기가 어렵다. 하지만 FCSO 알고리즘에서는 작은 클러스터로 나누어 클러스터의 노드수를 50개 이하로 만들기 때문에, 이 상태에서는 길지 않은 시간 안에 정확한 해를 구할 수 있어 FCSO 알고리즘을 제시하게 되었다.

4. 실험결과 및 분석

실험은 현재 운송시스템을 통해 얻어진 차량들의 총 이동거리와 제시한 CRA 알고리즘과 FCSO 알고리즘을 통해 얻어진 차량들의 총 이동거리를 비교하여 제시한 알고리즘의 성과를 측정하고자 한다.

현재 운송시스템은 40여 개의 수도권 물류업체들이 담당하는 권역이 나누어져 있어 고객들을 서비스하고 관리하고 있다. 이 물류업체들이 담당하는 권역을 하나의 클러스터로 보고 각 물류업체가 요구량을 충족시키기 위해 차량을 운용할 때 차량운행 총거리를 구하였다. 이러한 방법으로 계산한 현재 운송시스템의 차량 총 운행거리는 57,277이다.

실험은 수도권 물류업체들의 현황을 기초로 하여 석산 40개소, 레미콘 공장 120개소, 폐기물 집합소 40개소로 구성하였으며 각 노드들의 위치(좌표)와 생산량, 배달량, 수집량, 차량 대수 등은 연간 현황에 기초하였으나 실험의 효율성을 위하여 약간 변형되었다. 본 장에서는 제시된 두 가지 알고리즘에 대하여 실험하고 비교 분석한다. 실험 컴퓨터는 Pentium(R) CPU 2.80GHz, 512MB RAM을 사용하였으며, OPL Studio 3.6.1, Visual C++ 6.0을 이용하였다.

실험은 동일한 데이터를 바탕으로 클러스터의 수만을 변경하여 <표 1>과 같이 5개의 서로 다른 예제를 생성하여 실험하였으며, 각 예제는 클러스터의 개수를 5, 8, 10, 14, 20개로 다양하게 구성하였다. 이렇게 구성한 이유는 클러스터의 개수를 어떻게 나누었을 때 차량 총거리를 최소화할 수 있는지를 알아보기 위해서이다.

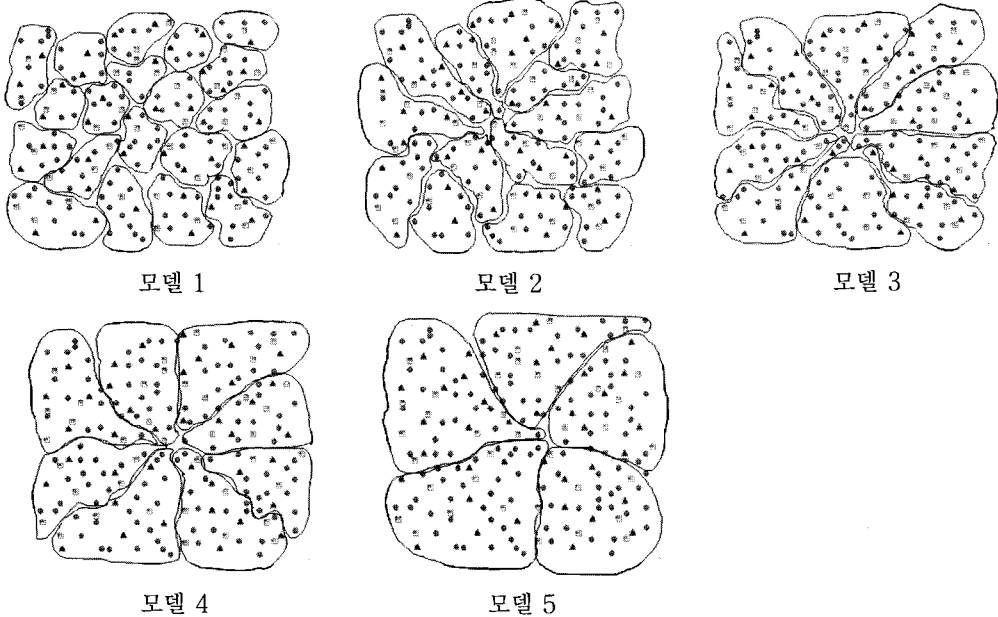
<표 1> 실험데이터

예제번호	클러스터 수	클러스터당 석산의 수
1	20	2
2	14	3
3	10	4
4	8	5
5	5	8

실험에서는 각 예제에 대하여 작은 클러스터로 나눈 후 각 클러스터에 대하여 최적경로를 구하여 차량 총거리를 구하고, 이를 모두 더하여 각 예제의 차량 총거리를 구하였다. 즉 석산들의 무게중심

으로 클러스터를 생성한 후에 CRA 방법에서는 차량경로를 구하고, 개선하는 방법으로 C++로 프로그램을 코딩하여 계산하였으며 FCSO 방법에서는 생성된 작은 클러스터에 대하여 OPL을 이용하여 차량경로를 결정하고 차량의 총 운행거리를 계산하였다.

현재 운용되고 있는 수송시스템의 차량의 총 이동거리는 57,277이며, <표 2>를 통해 CRA 방법의 경우 클러스터의 수가 10개일 때 29%의 운행거리 절감이 일어나며, FCSO 방법의 경우 클러스터의 수가 8개일 때 45%의 운행거리 절감이 일어남을 알 수 있다. FCSO 방법이 전체적으로 CRA 방법

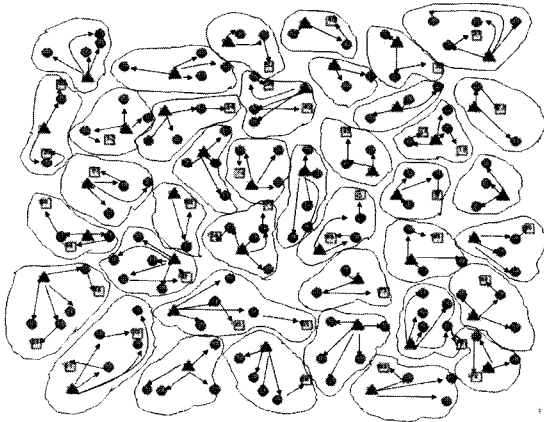


<그림 6> 실험데이터 예제유형

<표 2> 실험결과

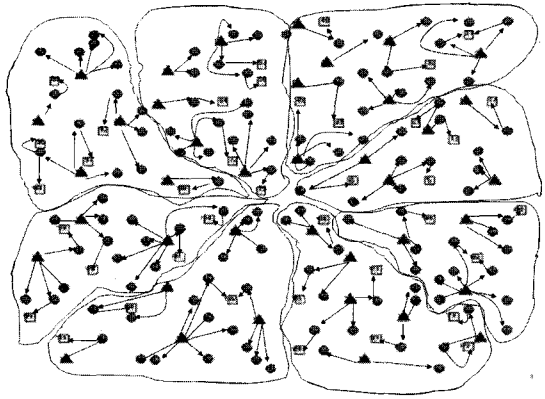
예제 번호	총 운행거리				계산 시간(초)	
	CRA	개선율(%)	FCSO	개선율(%)	CRA	FCSO
1	48,301	15	41,175	28	5	40
2	41,861	26	36,110	36	7	160
3	40,655	29	33,550	41	9	220
4	42,516	25	31,225	45	15	337
5	47,728	16	32,268	43	20	1,424

주) 개선율 = (현재 운송시스템 총거리 - 알고리즘 총거리) / 현재 운송시스템 총거리 * 100.



차량 운행 총거리 : 57,277

〈그림 7〉 현재 운송시스템



차량운행 총거리 : 31,225

〈그림 8〉 개선된 수송시스템

보다 더 좋은 값을 보였지만, CRA 방법이 계산시간 측면에서는 클러스터의 수가 늘어남에 따라 일정하게 증가하여 더 안정되며 실제 상황을 추가로 반영할 수 있다는 점에서 더욱 실용적이다.

이는 초기 클러스터를 구성할 때 무게 중심의 개념으로 묶어줌으로써 효과적으로 클러스터를 구성하여 시간을 절약시킬 수 있었고, 초기해를 개선할 때 UF 규칙에 의해 부적합한 고객들을 선별적으로 뽑아 이웃해를 생성하는 효율적인 알고리즘 덕분이다.

5. 결 론

본 연구에서는 두 가지 임무를 수행하는 차량의 총 이동거리를 최소화시키는 차량의 최적경로를 찾는 데 목적을 두었으며, 이 문제를 해결하기 위하여 새로운 최적화 수리모형을 제시하였다. 하지만 실제문제의 큰 사이즈 문제에 대해서는 최적화 수리모형을 이용하기에는 시간이 너무 많이 소요되어 적절한 해를 찾기가 어렵다. 따라서 이를 효과적으로 해결하기 위하여 CRA 알고리즘과 FCSO 알고리즘을 제시하였는데, CRA 알고리즘은 큰 사이즈의 문제를 무게중심 개념을 이용하여 작은 클러스터로 나눈 후, 각 클러스터에 대하여 최적 경

로를 찾고, 각 클러스터에서는 UF 규칙으로 가장 부적합한 해를 찾아 인접 클러스터로 이동시키는 방법을 이용하여 해를 최대 29%의 거리를 개선하였다. FCSO 알고리즘은 무게중심을 이용하여 작은 클러스터로 세분하는 과정은 같지만 그 이후로 최적화 수리모형을 이용하여 ILOG사의 OPL을 통하여 최적해를 구한다는 점이 상이하다. 이 알고리즘을 통하여 실제 문제를 다양하게 클러스터링해서 실험 후 총거리를 비교하였으며, 이 중에서 가장 좋은 클러스터링 방법과 차량 경로를 제시하여 최대 45%의 거리를 절감시킬 수 있었다.

위에서 제시한 알고리즘의 첫 번째 장점은 첫 번째 단계에서 무게중심을 이용함으로써 클러스터를 할 때 무작위적인 Sweeping 방법이나 Petal 방법보다는 훨씬 효율적으로 클러스터를 구성할 수 있다는 점이다. 초기해 생성 시 실제 공급 가용량을 고려한 무게중심을 기준으로 클러스터에 노드를 포함시키므로 기존의 무작위적인 방법보다 최적해에 가깝도록 노드를 선택할 수 있고, 따라서 최적해에 근사한 초기해를 구하여 전체적인 계산 시간을 절감할 수 있다. 두 번째 장점은 UF 규칙에서 클러스터 내의 가장 부적합한 고객들을 선택하기 위해 무게중심과의 거리차, 각도차, 원점과의 거리를 고려하여 해의 개선과정에서 빠르게 최적

해로 접근할 수 있다는 점이다. 또한 UF 규칙의 각 요소에서의 가중치를 변경시킴으로써 각 고객들의 UF 지수가 변하게 하여 다양하게 부적합 고객이 선택되도록 알고리즘이 구성되었다는 점이다.

반면에 단점으로는 해의 개선과정에서 부적합 고객을 선택할 때 UF 규칙에 있는 세 가지 요소의 가중치를 일정한 비율로 증가 또는 감소시키는데 그 비율을 경험에 의하여 찾아냄으로써 많은 시간이 소요되었다는 점으로 효과적으로 비율을 찾을 수 있는 방법이 연구되어야겠다.

본 논문에서 제시한 알고리즘을 이용하여 수도권에 있는 각 전자재 물류업체들이 통합배차를 한다면 각 회사별로는 차량 총거리가 늘어나거나 줄어들 수 있겠지만 전체적인 차원에서는 총거리가 절감되어 비용을 줄일 수 있을 뿐 아니라, 국가적으로도 에너지가 절약되어 통합배차가 필요하다고 할 수 있다. 향후 연구방향으로는 이와 유사한 유형으로 볼 수 있는 회수물류 관련 정책이나 환경문제에 관련된 recycling, reverse logistics 문제에 대한 효과적인 해법을 제시하도록 연구되어야 하겠다.

참 고 문 헌

- [1] 강자영, 장희정, 강장하, 박성수, "동일경로 제약을 갖는 집배송 차량 경로 수립 문제의 최적화 해법", 「대한산업공학회지」, 제33권, 제1호(2007), pp.33-43.
- [2] 이원동, 지창훈, 이영훈, 안승범, "전자재 물류시장의 협업적 차량운송 관리에 관한 연구", 「한국SCM학회지」, 제17권, 제1호(2007), pp.65-77.
- [3] 장희정, 이경식, 최은정, 박성수, "시간제약과 하역장 용량제약이 있는 차량경로문제에 대한 타부탐색 알고리즘", 「한국경영과학회지」, 제30권, 제4호(2005), pp.45-60.
- [4] 정은용, 박양병, "배달과 수거가 혼합된 차량 경로 결정문제를 위한 유전 알고리즘의 개발", 「대한산업공학회지」, 제30권, 제4호(2004), pp.346-354.
- [5] Bianchessi, N. and G. Righini, "Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery," *Computers & Operations Research*, Vol.34, No.2(2007), pp.578-594.
- [6] Brandao, J., "A new tabu search algorithm for the vehicle routing problem with back-hauls," *European Journal of Operational Research*, Vol.173, No.2(2006), pp.540-555.
- [7] Danzig, G.B. and J.H. Ramser, "The Truck Dispatching Problem," *Management Science*, Vol.6, No.1(1959), pp.80-91.
- [8] Irnich, S., "A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles," *European Journal of Operational Research*, Vol.122, No.2(2000), pp.310-328.
- [9] Nagy, G. and S. Salhi, "Heuristic algorithm for single and multiple depot vehicle routing problems with pickups and deliveries," *European Journal of Operational Research*, Vol.162, No.1(2005), pp.126-141.
- [10] Oukil, A., H.B. Amor, J. Desrosiers, and H. E. Gueddari, "Stabilized column generation for highly degenerate multiple-depot vehicle scheduling problems," *Computers & Operations Research*, Vol.34, No.3(2007), pp.817-834.
- [11] Pisinger, D. and S. Ropke, "A general heuristic for vehicle routing problems," *Computers & Operations Research*, Vol.34, No.8(2007), pp.2403-2435.
- [12] Salhi, S. and M. Sari, "A multi-level composite heuristic for the multi-depot vehicle fleet mix problem," *European Journal of Operational Research*, Vol.103, No.1(1997), pp.95-112.
- [13] Toth, P. and D. Vigo, "A heuristic algorithm

for the symmetric and asymmetric vehicle routing problems with backhauls," *European Journal of Operational Research*, Vol. 113, No.1(2002), pp.487-512.

[14] Wade, A.C. and S. Salhi, "An investigation into a new class of vehicle routing problem with backhauls," *Omega*, Vol.30, No.6(2002), pp.479-487.