

# 한정 용량 차량 경로 탐색 문제에서 이분 시드 검출 법에 의한 발견적 해법

고준택  
인하대학교 대학원 정보공학과  
(chakande@eslab.inha.ac.kr)

유영훈  
인하대학교 대학원 정보공학과  
(yhyu@eslab.inha.ac.kr)

조근식  
인하대학교 IT공과대학 컴퓨터정보공학부  
(gsjo@inha.ac.kr)

본 연구에서는 한정 용량 차량 경로탐색 문제(CVRP, Capacitated Vehicle Routing Problem)에서 이분 시드 검출 방법(Bisection Seed Detection)을 이용한 휴리스틱 알고리즘을 제안하였다. 이 알고리즘은 3단계로 구성된다. 1단계에서는 improved sweep 알고리즘을 이용해서 초기 클러스터를 구성한다. 2단계에서는 1단계에서 얻은 각 클러스터에 대하여 이분 시드 검출 법을 이용해서 seed 노드를 선택하고, regret 값에 따라 각 경로에 고객 노드들을 삽입 함으로서 차량 이동 경로를 생성한다. 3단계에서는 tabu 탐색 방법과 노드 교환 알고리즘(node exchange algorithm)을 이용하여 2단계에서 얻어진 각 경로를 더욱 향상 시킨다. 본 논문의 실험에서는 제안된 휴리스틱이 비교적 빠른 시간 내에 최적 근사 값을 얻을 수 있음을 보였으며, 이는 빠른 실행 시간을 요구하는 실 업무에 유용하다.

논문접수일 : 2008년 11월 08일    논문수정일 : 2009년 01월 09일    게재확정일 : 2009년 01월 15일    교신저자 : 유영훈

## 1. 서론

차량 경로 문제(VRP, vehicle routing problem)는  $n$ 개의 수요 지에 서비스를 제공하기 위해 동일 형태의  $k$ 대 차량이 물류 창고에 있을 경우 최소비용 또는 최소거리로 운송 할 수 있는 경로를 구하는 문제이다. 이러한 문제에서는 적재량과 경유 거리라는 제약들로 인하여 한 대의 차량으로 물량을 공급 받는 고객들의 총 수요량은 차량의 적재량을 넘을 수 없다. 또한, 경로의 총 거리는 최대 허용거리 보다 클 수도 없다. 일반적으로 차량 경로 문제의 목적은 차량 운행에 사용되는 비용을 최소화 하는 것이다. 한정 용량 차량 경로탐색 문제(CVRP)

는 수리적으로 NP-hard 문제로서 경로의 구성이 외관원 문제(TSP)를 반복적으로 구하는 문제와 같으며 차량의 적재량과 고객 물량의 제약이 있는 경우에는 배낭 문제(Knapsack)를 동시에 고려하게 되므로 계산 량이 많고 해를 구하기가 어렵다. 그러므로 이러한 해법에서 중요한 것은 계산 량이며, 문제의 크기가 커질수록 계산 량이 지수적으로 증가하게 되므로 실제로 해를 제공하지 못하는 경우도 생긴다. 이러한 문제를 해결하기 위한 방법으로 크게 최적 해법(Optimization Technique)과 발견적 해법(Heuristic Method) 두 가지로 나눌 수 있다. 최적 해법은 수리 계획법(Mathematical Programming)에서 사용하는 알고리즘을 이용하

\* 이 연구에 참여한 연구자(의 일부)는 2단계 BK21 사업지원을 받았음.

여 차량 경로 문제를 푸는 방법으로 최적 해를 보장 해 줄 수 있다는 장점을 가진 반면 일반적으로 계산이 복잡하고 수행 속도가 느리다는 단점이 있다. 반면에 발견적 해법은 최적 해를 보장할 수 없지만 근사한 해를 최적 해법보다 빠른 시간에 구할 수 있다. 본 논문에서는 두 가지 방법 중에 최적해법에 의한 CVRP 해법을 제안한다. 왜냐하면 실 업무에서는 수행시간이 오래 걸리는 최적 값보다는, 비록 근사값을 구하더라도 수행 시간이 빠른 해법을 요구하기 때문이다.

잘 알려진 발견적 해법으로는 Clarke-Wright에 의해 발표된 거리 절감 해법(saving algorithm) (Clarke G. and Wright J.W., 1964)과 Gillet와 Miller에 의해 발표된 sweep 해법이 있으며 (Gillett, B. and L. Miller, 1974), 최근의 발견적 해법으로는 유전자 알고리즘(genetic algorithm)과 타부 탐색 알고리즘(tabu search algorithm) 등이 있다(Baker, B. M. and M. A. Ayechev, 2003; M. Gendreau, A. Hertz, and G. Laporte, 1994).

본 논문에서는, 하나의 물류 창고에서 각각의 고객에 수요량을 충족시키기 위해 차량을 운송하는 문제에서, 운행거리를 최소화 할 수 있는 새로운 해법을 제시 하고자 한다. 이를 위해 정수계획법을 이용한 차량 경로 모형을 설정하였다. 설정된 모형이 NP-hard의 형태이므로 계산적으로 해를 찾는 것은 불가능하다. 따라서 기존의 CVRP 해결 방법을 개선 시킨 새로운 발견적 해법을 개발하여 차량 경로 모형의 해를 구하고자 한다.

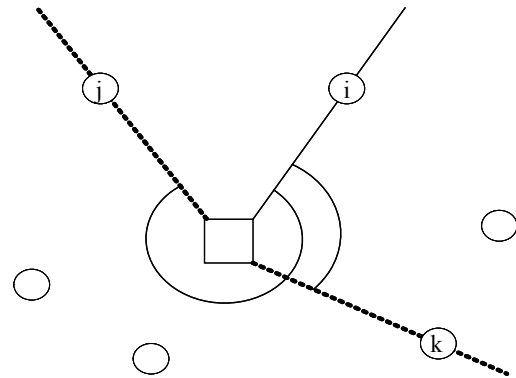
본 논문의 구성은 다음과 같다. 제 2장에서는 차량 경로 문제를 정의하고 관련 알고리즘을 소개한다. 제 3장에서는 본 연구에 대한 차량 경로 모형을 설정 하고, 제 4장에서는 본 연구에서 제안한 알고리즘에 대해 설명한다. 그리고 제 5장에서는 실험을 통해 제안한 알고리즘의 결과와 성능을 평

가하였다. 마지막으로 제 6장에서는 결론을 맺는다.

## 2. 관련 연구

### 2.1 한정 용량 차량 경로탐색 문제(CVRP)

한정 용량 차량 경로탐색 문제(CVRP, Capacitated Vehicle Routing Problem)는 가장 일반적인 차량 경로탐색 문제(VRP, Vehicle Routing Problem)가 확장된 형태이며 다음과 같이 설명될 수 있다. 하나의 물류 창고가 있고 그것을 depot = 0 이라 한다. 물류 창고에서는 용적량  $C$ 를 가진  $K$ 대의 독립된 차량을 운행하여 수요량  $D_i$ 를 가진  $n$ 명의 고객에게 서비스를 해야 한다( $i = 1, \dots, n$ ). 각각의 차량은 거리 비용을 최소화하는 경로를 구성해야 하며 여기서 비용  $C_{ij}$ 는 고객  $i$ 에서 고객  $j$ 까지의 거리이다( $i, j \in [1, n]$ ). 그리고 고객들간의 거리는 대칭적( $C_{ij} = C_{ji}$ )이며 또한  $C_{ii}$ 는 0이다. CVRP에 대한 해를 구하는 방법은 기본적으로 다음과 같다.  $K$ 개의 경로에  $n$ 명의 고객을 분배해서 할당한다. 고객을 할당하는데 있어 경로  $R_q$ 는  $\sum_{p \in R_q} D_p < C$  ( $p$ 는 경로  $R_q$ 에 속한 노드)과 같은 제약 조건을 만족해야 한다.



<그림 1> Sweep Algorithm

## 2.2 Sweep 알고리즘

<그림 1>에 Sweep 알고리즘은 VRP을 평면에 적용 시킨 경우이다. 초기의 경로들은 중앙에 있는 물류 창고에서 임의의 노드에 연결선을 그어 그 연결선을 회전시켜 나가면서 노드들을 경로에 할당함으로써 클러스터를 형성한다. 클러스터 형성 후 차량들은 각 클러스터에 대한 TSP문제를 풀으로써 해를 구하게 되고, 근접 경로들 간의 고객 노드 교환을 통해 해를 향상시킨다. 이러한 방법은 Gillet and Miller에 의해서 일반화 되었다(Gillett, B. and L. Miller. 1974). 이 방법의 절차는 다음과 같다.

- [단계 1] 고객 노드  $i$ 를 선택한다. 그리고 이 고객 노드를 포함하는 경로를 구성한다.
- [단계 2] 고객 노드  $i$ 와 창고, 그리고 시계 반대 방향에 있는 고객 노드  $j$ 간의 각  $\theta_j$ 를 구하고  $\theta$ 가 증가하는 순서대로 고객 노드들을 정렬시킨다.
- [단계 3] 정렬 순서에 따라서 고객 노드를 선택한다. 선택된 경로에 추가가 가능하다면 추가시키고 그렇지 않다면 새로운 경로를 시작해서 그 경로에 고객 노드를 추가시킨다.
- [단계 4] 경로에 할당되지 않은 고객이 없을 때까지 [단계 2]를 반복한다.
- [단계 5] 구해진 각 경로에 대해 TSP문제를 푼다.

## 2.3 Clarke-Wright 알고리즘

Clarke-Wright 알고리즘은 가장 잘 알려진 발견적 해법이다(Clarke G. and J. W. Wright, 1964). 이 방법의 주된 아이디어는 절약 개념을 바탕으로

초기해로 주어진 단독 경로들을 합쳐 나가는 것이다. 즉 이들 단일 경로에 대해 두 경로를 하나로 연결 시킴으로써, 차량을 줄이고 비용도 줄이게 된다. 두 개의 수요 노드  $i$ 와  $j$ 를 두 대의 차량으로 공급 받는다고 하자, 노드  $i$ 와 노드  $j$ 를 연결하여 두 대의 차량 대신 한대의 차량으로 운행 한다면 운행거리에 있어 다음과 같은 절약 값이 발생된다.

$$(d_{si} + d_{is} + d_{sj} + d_{js}) - (d_{si} + d_{ij} + d_{js}) = (d_{is} + d_{sj}) - d_{ij}$$

여기서  $s$ 는 출발지를 의미하고,  $d_{ij}$ 는 노드  $i$ 와 노드  $j$ 사이의 거리를 의미한다. 따라서 두 개의 경로를 연결하면 위에서 구한 두 노드의 절약된 거리만큼 이동거리가 단축되게 된다. Clarke-Wright 방법의 해를 구하는 과정은 다음과 같다.

### [단계 1] 초기 해 생성

- 모든 고객에 대해 거점( $s$ )-고객노드( $i$ )-거점( $s$ ) 형태의 초기해  $S$ 를 생성한다.

### [단계 2] 절약 비용 계산

- 모든 경로에 대해 그 두 경로를 하나의 경로로 합치는 경우의 절약 비용(saving cost)을 계산한다.

### [단계 3] 절약효과 발생 유무 확인

- 0보다 큰 절약 비용이 존재하지 않으면 [단계 5]로 가고 그렇지 않은 경우 [단계 4]로 간다.

### [단계 4] 두 경로의(link) 병합

- 0보다 큰 절약 비용 중에서 CVRP제약 조건을 만족하는 경우가 없으면 [단계 5]로 가고, 그렇지 않으면 모든 제약조건을 만족하면서 가장 절약 비용이 큰 경로를 하나의 경로로 합친 후 [단계 2]로 간다.

### [단계 5] 알고리즘 종료

## 2.4 유전자 알고리즘

유전자 알고리즘(genetic algorithm)은 어려운 최적화 문제를 해결 하는데 있어서 일반적으로 많이 이용되는 메타 휴리스틱(meta-heuristic) 방법이다. 유전자 알고리즘은 적자생존과 자연도태의 진화원리를 컴퓨터를 이용해 구현하는 방법이다. 유전자 알고리즘은 확률적인 탐색을 통해 어렵고 복잡한 문제에 대해 근사 최적해(near optimum solution)를 제공하는 발견적 해법이다. 유전자 알고리즘은 복잡한 해공간(solution space)에서 강력한 탐색능력을 가지고 있으며 일반적으로 적은 시간과 비용이 소요된다는 장점이 있다. 유전자 알고리즘은 생물학적인 구조의 유전자를 표현하기 위해 이진정수나 실수를 이용하여 문자열을 구성한다. 이를 통해 자연계의 진화과정인 선택(selection), 교차(crossover), 그리고 돌연변이(mutation)에 의해 주어진 문제에 대해 최적 점에 근접하는 개체를 생성하여 최적 해를 찾아가는 절차를 수행한다.

## 2.5 타부 탐색

타부 탐색(Tabu Search)은 인간의 기억 과정을 이용한 복잡한 해 영역에서 근사 최적 해를 찾기 위한 경험적 해법이다. 어려운 문제를 쉽게 다루고 많은 제약 사항을 동시에 고려 할 수 있으며 다른 휴리스틱 과정을 제어할 수 있는 메타 휴리스틱 중 하나인 타부 탐색은 F.Glover에 의해 현재의 형태로 발전 되었다(Gendreau, M., A. Hertz, and G. Laporte, 1994). 타부 탐색은 언덕 오르기 휴리스틱(Hill-climbing heuristic)과 유사한 면이 많다. 우선 언덕 오르기 휴리스틱 기법에 대해 살펴보면 다음과 같다.

단계 1 : 초기값  $s_0 \in X$  을 선택한다.

단계 2 :  $c(s) < c(s_0)$ 를 만족하는 어떤  $s \in S(X)$ 를 선택한다.

단계 3 :  $s_0 = s$ 라 둔다, 단계 2로 돌아간다.

여기서  $X$ 는 해의 공간이고,  $S(X)$ 는 이웃해 집합이다. 즉, 초기해는  $s_0$ 를 현재해로 한 후 탐색을 시작하여 이웃해 집합을 구한다. 그리고 그 이웃해 중에서 최소화가 되는 이웃해이면 그 해를 현재해로 한 후 탐색을 계속하고 그렇지 않으면 끝낸다. 하지만 이 기법은 시작점에서 지역 최적해를 향하여 유일한 한쪽 방향으로만 진행하기 때문에 지역 최적 해를 얻기는 하나 전체 최적 해라는 보장이 없다는 한계점을 가진다. 이런 문제점을 보완 하기 위해서 타부 탐색은 통제에 관련된 기술로 메모리 구조를 이용한 타부 목록과 열망수준을 사용하여 자유롭게 탐색 함으로써 지역 최적 해에 머무르는 단점을 보완하고 있다. 타부 목록(tabu list)이란 일반적으로 해의 이동정보를 나타내는 타부 속성을 일정기간 동안 기억하는 것으로, 최근에 생성된 한정된 해의 이동정보를 타부 목록에 기억하는 단기 기억(short term memory)과 처음부터 현재까지를 기억하는 장기기억(long term memory)을 이용하는 것 등이 있다. 본 논문에서는 단기 기억 타부 탐색 기법을 이용한 것으로 타부 목록의 크기를 정해주어야 한다. 타부 목록 크기는 타부 속성의 개수를 나타낸 것으로써 정해진 개수만큼 타부 속성이 채워지고 새로운 타부 속성이 들어오면 기존의 타부 목록에 있던 속성 중 가장 먼저 들어왔던 타부속성이 빠져나가는 선입선출(First In First Out) 방식으로 변화한다, 그러나 이러한 타부 목록의 사용이 오히려 좋은 해가 있을 수 있는 곳으로 이동하는 것을 방해할 수 있다. 이런 한계를 극복하기 위해서 어떤 때에는 타부 목

록을 무시할 수 있는 새로운 조건이 필요한데 이것을 열망수준(aspiration criteria)이라고 한다. 열망수준은 현재의 해가 타부 상태에 있으나 해가 어느 수준 이상이면 해의 이동을 허락하는 기준으로 사용 되는데, 일반적으로 현재까지 발견된 가장 좋은 해를 열망수준으로 이용한다.

### 3. 차량 경로 모형

#### 3.1 문제 정의

차량 경로 모형에 대한 문제 정의는 다음과 같다.

- 1) 모든 차량은 동종의 차량이다.
- 2) 운행되는 차량의 비용은 총 운행 거리에 비례 한다.
- 3) 각 차량은 최대 적재용량까지 적재가 가능하며 모든 차량의 적재 용량은 동일하다.
- 4) 고객 노드들 간의 차량 운행속도는 동일하다 (적재용량에 영향을 받지 않는다).
- 5) 화물의 적재 및 하역에 소요되는 시간은 고려 되지 않는다.
- 6) 차량의 출발 및 종착은 물류 창고에서만 이루어 지며, 단일 물류 창고만 존재한다.
- 7) 모든 고객 노드들은 서비스를 받아야 한다.
- 8) 차량이 경로를 운행하는 거리에 대한 제약은 고려 하지 않는다.

#### 3.2 변수 정의

차량 경로 모형에서 사용되는 변수들의 정의는 다음과 같다.

$N$  : 고객 노드  $i$ 의 집합,  $i = 1, \dots, I$   
 $N'$  : 후보 seed 고객 노드  $j$ 의 집합,  $j = 1, \dots, J$

$K$  : 가용 차량의 집합,  $k = 1, \dots, K$   
 $R$  : 생성된 경로의 집합,  $r = 1, \dots, R$   
 $Q_k$  : 차량  $k$ 에 의해 서비스 되는 고객 노드의 집합  
 $d_{ij}$  : 구간( $i, j$ )의 거리  
 $D_i$  : 고객 노드  $i$ 의 수요량  
 $C$  : 차량의 적재 용량  
 $A_{ij}$  : 만일 고객 노드  $i$ 가 seed  $j$ 의 집단에 속해 있다면  $A_{ij} = 1$ , 그렇지 않다면  $A_{ij} = 0$ 이다. 변수  $A$ 는 할당 변수이다.  
 $S_j$  : 만일 고객 노드  $j$ 가 seed라면  $S_j = 1$ , 그렇지 않다면  $S_j = 0$ 이다. 변수  $S$ 는 후보 seed 노드  $j$ 가 seed인지 아닌지 나타내는 변수이다.

#### 3.3 제약 조건 식과 목적 함수

CVRP 모형은 운행하는 차량의 수가 정수이기 때문에 차량의 운행 구간 수는 반드시 정수가 되어야 한다. 따라서 정수계획법을 이용하여 모든 화물을 배송시키면서 운행거리를 감소시켜 운행 비용을 최소화하는 모형을 구성하였다. 본 연구에 영향을 주는 제약 조건 식은 다음과 같다.

경로에 대한 수요량의 총합이 차량의 적재 용량을 초과하지 않도록 해야 한다. 차량은 적재 용량을 초과하지 않는 범위에서 최대한 적재한다. 화물을 적재하고 구간( $i, j$ )을 운행하는 차량의 적재 용량  $C$ 는 경로 내의 수요량의 합보다 크거나 같아야 한다. 이러한 조건은 식 (1)과 같이 표현된다.

$$\sum_{i \in N} d_i A_{ij} \leq C, \quad \forall j \in N' \quad (1)$$

경로에 대해 고객노드들의 중복 할당을 방지하기 위해 다음과 같은 제약 조건이 필요하며, 식 (2)과 같이 표현된다

$$\sum_{i \in N} R_i = 1, \quad \forall i \in N \quad (2)$$

seed에 대해서도 고객노드들의 중복 할당을 방지 하기 위해 다음과 같은 제약 조건이 필요하며, 식 (3)과 같이 표현된다.

$$\sum_{j \in N'} A_{ij} = 1, \quad \forall i \in N \quad (3)$$

seed로 선택되지 않은 후보 seed  $j$ 에 고객 노드  $i$ 가 할당 되는 것을 방지 하는 제약 조건은 식 (4)과 같이 표현된다.

$$A_{ij} \leq S_j, \quad \forall i \in j, j \in N' \quad (4)$$

끝으로, 정확히  $K$ 개의 seed들이 선택되어야 한다는 제약 조건이 있으며 식 (5)과 같이 표현할 수 있다(이용 가능한 차량의 수만큼).

$$\sum_{j \in N'} S_j = K \quad (5)$$

본 모형의 목적함수는 화물을 운송하는데 필요한 모든 차량의 총 운행 비용(거리)  $V$ 를 최소화시키는 것이다. 본 모형의 목적 함수는 식 (6)과 같이 표현된다.

$$\text{Minimize } V(A, S) = \sum_{i \in A} d_{ij} A_{ij} \quad (6)$$

따라서, 차량 경로 모형의 최종 목적 함수는 화물을 운송하는데 필요한 모든 차량의 총 운행비용  $V$ 를 최소화 시키는 것이다. 본 논문의 차량 경로 모형은 다음과 같이 표현될 수 있다.

$$\begin{aligned} \text{minimize } V &= \sum_{i \in N} \sum_{j \in N'} d_{ij} A_{ij} \\ \text{subject to : } &\sum_{i \in A} d_{ij} A_{ij} \leq C \quad \forall j \in N' \\ &\sum_{i \in N} R_i = 1, \quad \forall i \in N \end{aligned}$$

$$\begin{aligned} \sum_{j \in N'} A_{ij} &= 1 \quad \forall i \in N \\ A_{ij} &\leq S_j \quad \forall i \in N, j \in N' \\ \sum_{j \in N'} S_j &= K \\ (A_{ij}, S_j) &= (0, 1) \end{aligned}$$

## 4. 이분 시드 검출 법

본 논문에서 제안하는 발견적 해법은 3단계의 절차로 구성된다. 1단계에서는 sweep 알고리즘을 확장한 improved sweep algorithm을 이용해서 초기 클러스터를 구하게 되고, 2단계에서는 1단계에서 구한 각 경로에 seed 노드를 설정하고, regret 함수를 이용하여 고객 노드들을 seed 노드에 할당함으로써 더 좋은 클러스터를 구성하게 되며, 3단계에서는 각 경로에 tabu search를 적용해 라우트를 구성하게 되고, 마지막으로 노드 교환 알고리즘을 통해 더욱 결과를 향상 시킨다.

### 4.1 1-단계 절차

1단계는 초기 경로를 구성하기 위한 클러스터 생성 단계이다. 이 과정에서의 초기 클러스터 구성은 매우 중요하다. 초기 클러스터가 얼마나 좋은가에 따라서 최종 결과에 큰 영향을 미친다. 기존의 sweep 알고리즘은 단순히 각으로써 클러스터와 경로를 동시에 구성 했기 때문에 경로를 구성하기 위한 시작 노드가 어디인가에 따라 초기 경로 값이 달라져 최적 초기 경로를 구성 하는데 어려움이 있었다. 본 논문에서는 sweep 알고리즘을 확장한 improved sweep 알고리즘을 적용해서 나오는 여러 해들 가운데 가장 좋은 결과를 초기 경로로 한다. 이와 같은 방법을 사용하면 기존 sweep 방법을 사용하는 것보다 좋은 초기 해를 가지고 시작함으로써 최선 해를 구하는데 드는 수행 시간

을 단축시킬 수 있다. 1단계에서의 초기 클러스터와 경로를 구하는 과정은 다음과 같다.

#### First Phase Procedure

- (1) Until  $N = \emptyset$
- (2) StartNode=RandomSelectCustomer(N, S)
- (3)  $N = N - S$
- (4) ComputeAngle( $\theta_{Si}$ )
- (5) Sort( $\theta_{Si}$ )
- (6) While  $N \neq \emptyset$
- (7) If  $\text{sumDemandRoute}(Di) < C$
- (8) Then insertRoute( $\theta_{Si}, i$ )
- (9) else newRouteCreate(R)
- (10) insertRoute(R, i)
- (11) end while
- (12) If FindResult < Result
- (13) then Result = FindResult
- (14) end until

First Phase Procedure는 until과 while문에 의한 반복으로 구성된다. until문의 종료 조건은 고객 노드 집합(N)이 공집합이 될 때까지 반복되며, 고객 노드 집합 N에서 임의의 고객 노드를 선택해서 시작점으로 하고, 한번 선택된 노드는 고객 노드 집합 N에서 제외시킨다. (4)에서는 각  $\theta_{Si}$ 을 계산 한다. 여기서 각  $\theta_{Si}$ 는 시작 노드와 물류 창고, 그리고 고객 노드  $i(i \in N)$ 가 이루는 각이다. (5)에서는  $\theta_{Si}$ 를 오름차순으로 정렬한다. 정렬된  $\theta$ 에 따라 고객 노드들을 경로에 할당하는 과정이다. (6)에 나온 if문은 구성된 경로에 있는 고객 노드의 수요 합이 차량의 capacity보다 작다면 경로에  $\theta_{Si}$ 가 작은 순서대로 노드를 추가하며, 그렇지 않다면 새로운 경로를 만들어 그 경로에 고객 노드를 추가한다. 각 경로들에 할당되지 않은 노드가 더 이

상 없을 때까지 while문을 반복한다. (12)에서는 기존 해와 새로 구해진 해를 비교하여 새로 구한 해가 기존의 해보다 더 좋으면 그 결과 값을 해로 놓는다. 위와 같은 반복과정을 통해 총 운행 비용이 가장 적게 드는 경로를 초기 클러스터와 경로를 생성한다.

#### 4.2 2-단계 절차

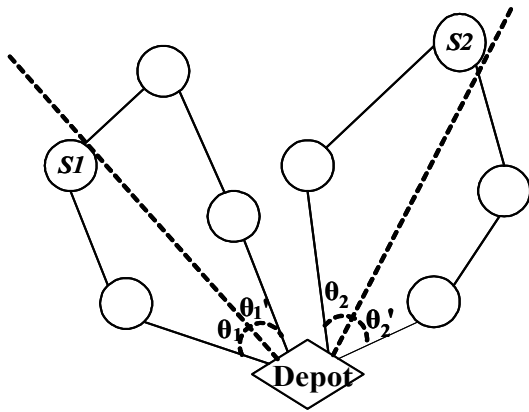
2단계는 1단계의 초기 클러스터를 바탕으로 경로를 재 구성 한 다음 더 좋은 경로를 생성하기 위한 과정이다. 2단계 휴리스틱 과정은 다음과 같다. 2단계 과정에서는 수행 시간을 단축시키기 위해 seed를 선택하기 위한 과정에서 이분 시드 검출법(Bisection seed detection)을 통해 seed의 자격을 갖춘 노드를 쉽게 찾아냄으로써, seed를 선정하는데 걸리는 수행 시간을 감소시켰다.

#### Second Phase Heuristic

Call one-phase heuristic

- (1) selectSeed(R, N, S)
- (2) insertSeed(seedSet, S)
- (3) deleteCustomer(N, S)
- (4) result = result of phase I
- (5) Until  $N = \emptyset$
- (6) While  $N \neq \emptyset$
- (7) RandomSelectCustomer(N, i)
- (8) regretValue=computeRegretValue(S1, S2, N, i)
- (9)  $N = N - i$
- (10) regretValueSort(i)
- (11) Seed= nearSeedSort(Seedk, i)

- (12) end while
- (13) While seedSet  $\neq \emptyset$
- (14) if sumDemandRoute(Rk, Di, sorted i) < C
- (15) then insertRoute(Rk, nearSeed)
- (16) seedSet = seedSet - nearSeed
- (17) end while
- (18) if  $i \notin Rk$
- (19) then createNewSeed(S, i)
- (20) seedSet = seedSet + i
- (21) end until



<그림 2> Seed노드 선택 방법

Second Phase Heuristic의 (1)은 seed값을 선택하는 단계이다. 1단계에서 구한 경로들을 초기 경로로 하여 seed값을 구한다. <그림 2>에서는 seed값을 구하는 방법을 보여준다. 각 경로에 이등분선을 그어 그 선에 가장 가까운 각(degree)을 가진 고객 노드를 경로의 seed(S1, S2)로 설정한다. 이와 같은 방법으로 seed를 선택하는 첫 번째 이유는, 1단계에서 경로를 구하는데 중요한 요소로 각도를 사용했기 때문이다. 각 경로의 중심 값(seed)을 설정하고 고객들을 seed에 할당 하는데 있어 다른 요소들 보다 각을 기준으로 seed를 선

택하는 것이 더 유리 할 것이다. 두 번째로는 기존의 다른 연구 들에 비해 효율적이면서도 간단한 과정을 통해 seed 를 선택함으로써, 해를 구하는데 총 수행 시간을 감소 시키기 위함이다.

(5)~(21)까지는 until문에 의한 반복으로 이 과정에서 향상된 경로를 구하게 된다. Until문 안에 두 개의 while문을 이용해서 첫 번째 while문(6~12)에서는 고객 노드 집합 N에서 랜덤으로 선택된 노드 i에 대한 regret 값을 구한다. regret 값을 구하는 방법은 다음의 식 (7)과 같다.

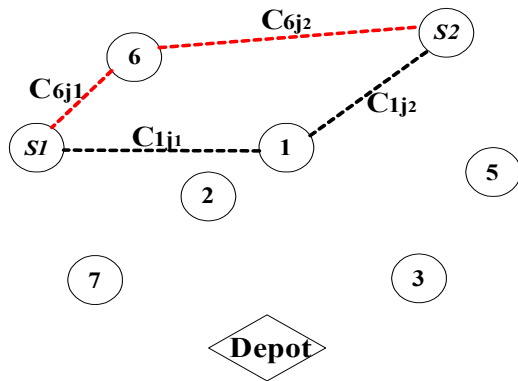
$$REGRET(i) = C_{ij2} - C_{ij1} \quad (7)$$

고객 노드 i와 가장 가까운 seed  $C_{j1}$ 와 두 번째로 가까운 seed  $C_{j2}$ 와의 차를 나타내는 것이 regret 값이다. regret 값의 의미는 고객 i를 두 번째로 가까운 seed에 할당하면, 첫 번째로 가까운 seed에 할당 했을 때 보다 손해 보는 비용을 나타낸다. 만일, 이 값이 크다면 이 고객 먼저 가장 가까운 seed 노드에 할당하는 것이 총 비용을 감소 시키는데 가장 좋을 것이다. 선택된 고객 노드 i는 고객 노드 집합 N에서 삭제한다. 그리고 노드 i의 값을 정렬하고 i와 다른 seed 간의 거리를 계산한다. 이와 같은 과정을 고객 집합 안의 모든 고객들이 선택될 때까지 계속 반복한다. 이 과정을 통해서 고객들은 regret 값에 의해 정렬 되고, regret 값을 계산하기 위한 S1과 S2 구할 수 있다. (13)의 while문은 regret값에 의해 sorting된 고객 노드 i를 경로에 할당하는 과정이다. (14)~(16)에서는 경로  $R_k$ 가 차량의 수용량 C를 넘지 않는다면, 그 경로에 regret 값이 가장 큰 고객 노드 i를 먼저 경로에 할당한다. seed에 할당된 고객 노드 i는 노드 집합에서 삭제하고 동일한 seed에 고객이 두 번 할당되는 것을 방지 한다.



<표 1> 고객의 좌표 데이터

Node no.	X Coordinate	Y Coordinate
Depot	0	0
1	1	6
2	-2	4
3	3	2
4	4	9
5	5	5
6	-3	8
7	-4	2
8	-5	6



<그림 3> Regret 함수 값 구하는 방법

(18)~(20)에서는 만일 고객  $i$ 가 경로들의 제약 조건 때문에 어디에도 할당 되지 못했다면  $i$ 를 새로운 seed로 설정하고, seed집합에  $i$ 를 추가 시킨다. 그리고 나서 다시 util문을 반복해서 최선의 경로를 구성하게 된다. 위의 휴리스틱 과정을 더 쉽게 설명하기 위해, <표 1>에서 랜덤으로 발생시킨 좌표를 예로 들어 휴리스틱 과정을 설명하였다.

다음은 <그림 2>와 <그림 3>에서 나타낸 seed 노드 선택 방법을 이용하여 구한 seed들에 대해 고객 노드 1과 6의 regret 값을 구하는 방법을 보여준다.

$$\text{노드 1의 regret 값 : REGRET}(1) = C_{1j2} - C_{1j1}$$

$$\text{노드 6의 regret 값 : REGRET}(6) = C_{6j2} - C_{6j1}$$

다른 노드들도 동일한 방법으로 regret 값을 구할 수 있다. <표 2>는 각 고객 노드들에 대해 regret 값을 구해서 seed들에 대해 고객을 할당할 순서를 정하고, 어떤 seed에 삽입 할 것인지를 보여준다. Regret 값에 따라 경로에 대한 노드 삽입 순서가 정해진다. 아래의 <표 2>에서는 고객 7의 regret 값이 가장 크기 때문에, 가장 먼저 고객 7과 가장 가까운 seed에 할당되고, 나머지 고객들도 이와 같은 방법으로 각 seed에 할당하게 된다.

<표 2> Regret 함수 값에 따른 노드의 정렬과 삽입

Cus no.	Regret Value	Insertion Order	Nearest Seed
1	1.76	6	Seed2
2	4.20	4	Seed1
3	1.87	5	Seed2
4(Seed1)	-	-	-
5	5.93	2	Seed2
6	4.24	3	Seed1
7	6.51	1	Seed1
8(Seed2)	-	-	-

### 4.3 3-단계 절차

3단계에서는 2단계에서 새롭게 구해진 경로에 대해 단기 메모리를 이용한 타부 탐색을 적용해서 각 경로들에 라우팅 문제를 풀었다. 마지막으로 경로를 향상시키기 위한 알고리즘으로 노드 교환 휴리스틱(Exchange heuristic)을 사용했다(Stefan, R., 2005). 고객 노드들에 대한 교환은 연결된 모든 고객 노드들을 교환해보는  $n:n$  교환이 아닌, 교환 휴리스틱에서 가장 많이 사용되고 효율적인

1 : 1 교환 알고리즘을 이용했다(Stefan, R., 2005).

## 5. 실험 및 평가

본 연구에서 제시한 CVRP를 위한 발견적 해법에 대한 효율성을 검증하기 위하여, improved sweep algorithm과 가장 많이 사용되는 발견적 기법 중 하나인 Clarke-Wright 휴리스틱(CW), 메타휴리스틱 중 하나인 유전자 알고리즘(GA, Genetic Algorithm)과 비교했다. 실험 환경은 Pentium IV 3.2GHz, 1GB RAM의 PC이다. <표 3>은 Christofides, Mingozz and Toth의해 만들어진 실험 데이터 집합과 그에 대한 알려진 최적 값을 보여주고 있다(Taillard E. D., 1993; Christofides, N., A. Mingozzi, P. Toth, 1979; Gendreau, M., Hertz, A., and Laporte, G., 1994; Osman, I. H., Netastrategy, 1993). 공정한 실험 결과의 비교를 위해 이와 동일한 데이터를 사용하였다.

<표 3> Test data

Prob.	N	Q	L	$\delta$	Best Published solution
C1	50	160	$\infty$	0	524.61
C2	75	140	$\infty$	0	835.26
C3	100	200	$\infty$	0	826.14
C4	120	200	$\infty$	0	1042.11
C5	150	200	$\infty$	0	1028.42
C6	199	200	$\infty$	0	1291.29

주)  $N$  : total number of customer.  
 $\delta$  : service time for a customer.  
 $Q$  : capacity of vehicle.  
 $L$  : route length constraints of vehicle.

<표 4>에서, 총 운행 비용을 비교해 볼 때 제안된 발견적 해법은 sweep이나 Clarke-Wright방법

보다는 좋은 결과를 산출해냈다. 하지만 GA과 비교해 본다면, 고객 수가 작을 때에는 GA보다 더 좋은 결과를 생성 했지만, 고객의 수가 많아질 수록 GA보다 총 운행 거리가 좋지 않았다. 이와 같은 결과는 본 논문에서 제안한 이분 시드 검출법이 seed를 한번에 선택 함으로써 총 수행 시간은 감소 시키지만, 경로를 구성 하는데 있어서 좁은 각도(degree)를 가진 경로에 많은 고객들이 포함 되어 있는 경우에는 이분 선에 근접한 고객들이 많아져서 효율적인 seed 선택이 되지 못하기 때문인 것으로 분석되었다.

<표 4> Heuristic간의 총 운행 거리 비교

Prob.	Size	CW	GA	Our best Solution
C1	50	585.00	526.74	524.61
C2	75	900.00	860.64	850.17
C3	100	886.00	849.30	859.72
C4	120	1079	1049.64	1069.25
C5	150	1204.0	1069.93	1078.73
C6	199	1504.0	1335.87	1349.87

<표 5>는 휴리스틱 방법들 중, 결과 값이 비슷한 GA방법과 본 논문에서 제안한 이분 시드 검출법을 사용한 방법, 그리고 최적 값과의 비교이다. 최적 값과의 차이가 GA는 평균 2.33%, 시드 이분법은 평균 2.86% 차이를 보였다. 특히 본 논문에서 제시한 기법은 고객 노드의 개수가 50인 데이터에서 최적 값과 같은 값을 찾아 냈고, 고객 노드가 75개인 데이터에서도 GA보다 나은 결과를 보였다. 하지만 전체적인 평균 값으로 볼 때에는 GA보다 다소 떨어지나 고객 수가 적을 때는 더 좋은 결과가 나왔으므로, 구체적인 성능 평가를 위해 해를 구하기 위한 수행 시간을 <표 6>에서 추가하여 비교하였다.

<표 5> 최적 값과의 차이 비교

Prob.	Size	Best	Our best		GA	
			cost	diff %	cost	diff %
C1	50	524.61	524.61	0	526.74	0.37
C2	75	835.26	850.17	1.75	860.64	2.95
C3	100	826.14	859.72	3.91	849.30	2.73
C4	120	1042.11	1069.25	2.54	1049.64	0.72
C5	150	1028.42	1078.73	4.66	1069.93	3.88
C6	199	1291.29	1349.87	4.34	1335.87	3.34
Avg				2.86		2.33

<표 6>은 이분 시드 검출 법을 사용한 방법과 GA과의 수행 시간 비교를 보여주고 있다. 논문에서 제안한 방법은 초기 클러스터 과정에서 기존의 sweep 알고리즘을 개량시킨 improved sweep algorithm을 사용하여 최적 값에 가까운 근사 해를 산출했으며, 이를 바탕으로 최종 해를 도출하기까지의 수행시간을 향상 시켰다. 그리고 seed를 선택하는데 있어서 이분법을 사용함으로써 기존의 seed 선택 방법(Christofides, N., A. Mingozzi, P. Toth, 1979)보다 빠른 시간 안에 최적 경로를 구성할 수 있는 seed를 선택할 수 있도록 했다. 또한 타부 탐색(tabu search)을 이용해서 기억장소의 절약과 수행시간을 단축시켰다. 이와 같은 방법을 이용해서 최적 값과 근사한 최종 해를 도출하는데 있어서 다른 휴리스틱보다 빠른 시간 안에 최적 해와 근사한 해를 도출 할 수 있었다. 유전자 알고리즘(GA)은 사용하는 매개 변수 값이 해의 수렴 속도에 많은 영향을 미치므로, 다양한 수준의 매개변수들을 수행하여 가장 우수한 결과를 가져 오는 조합의 수준을 선택하면 된다. 실험을 통해 최적 환경으로 결정된 매개변수는 다음과 같다. 초기 해의 개수는 고객의 수로 정하고, 교배율을 0.1, 역전율을 0.3, 돌연변이를 0.05으로 각각 적용하였다. 종

료 조건으로는 해의 향상 없이 5회 이상 반복된다면 더 이상 진화가 없는 것으로 판단하고 종료했다. 제안한 알고리즘과 GA의 수행 시간을 실험해본 결과, GA의 단점 중 하나인 변수의 제한조건을 최적화 했음에도 불구하고, <표 6>과 같이 제안한 알고리즘이 GA보다 모든 데이터의 수행시간 측면에서 월등히 좋은 것을 확인 할 수 있었다.

<표 6> 유전자 알고리즘과의 수행 시간 비교

Prob.	Size	Our best		GA	
		cost	time(s)	cost	time(s)
C1	50	524.61	0.11	526.74	1.92
C2	75	850.17	0.89	860.64	18.01
C3	100	859.72	1.42	849.30	32.40
C4	120	1069.25	3.45	1049.64	70.26
C5	150	1078.73	9.87	1069.93	141.61
C6	199	1349.87	33.52	1335.87	579.98

## 6. 결론

본 연구에서 다룬 한정 용량 차량 경로탐색 문제의 목적은 차량의 이동 거리 최소화와, 최적에 근접한 해를 빠른 시간 안에 구하는 것이었다. 이를 위해 이분 시드 검출 법을 이용하여 차량경로 문제에 대한 발견적 해법을 제시하였다. 이 방법은 3단계로 이루어져 있으며, 첫 단계에서는 기존의 sweep알고리즘을 개선시킨 improved sweep알고리즘을 사용해서 최선의 초기 클러스터를 형성했고, 2단계에서는 단계 1의 클러스터를 개선시키기 위해 이분 시드 검출 법을 사용해서 클러스터와 경로를 재구성하였다. 3단계에서는 타부 탐색과 노드 교환 알고리즘을 이용해서 최종 결과를 이끌어 내었다. 제안한 기법의 효율성을 검증하기 위해

가장 많이 사용되는 발견적 해법인 CW 휴리스틱과 메타 휴리스틱 중 하나인 유전자 알고리즘과 비교하였다. 차량의 총 운행비용 측면에서는 다른 기법과 동일하거나 우수했으며, 수행 시간 측면에서는 비교한 기법에 비해 월등히 좋은 것을 확인할 수 있었다. 따라서 본 연구에서 제시한 발견적 해법은 비록 최적 값이 아닌 근사 값이더라도 빠른 시간 내에 결과를 얻어서 즉시 사용할 수 있기에 실시간 대응을 요구하는 산업 현장의 요구에 더 유용한 해법이라고 할 수 있다. 그러나 산업 현장의 라우팅 환경은 시간 제약 조건을 비롯하여 다양한 제약조건들이 존재하기에 이를 반영하기 위한 추가 연구가 필요하다.

## 참고문헌

- Taillard, E. D. "Parrel iterative search methods for vehicle routing problem", *Networks*, Vol.23(1993), 661~673.
- Clarke G. and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points", *Operational Research*, Vol. 12(1964), 568~581.
- Koskosidis I. and W. B. Powell, "Clustering algorithms for consolidation of customer orders into vehicle shipments", *Transpn.Res.-B*. Vol.26B, No.5(1992), 365~379.
- Clarke, G. and J. W. Wright, "Scheduling of vehicles from a Central Depot to a Number of Delivery Points", *Operational Research*, Vol. 12(1964), 568~581.
- N.Christofides, Mingozzi, A., and P. Toth, "The vehicle routing Problem", In N.Christofides, A. Mingozzi, and C. Sandi, edit ors, *Combinatorial Optimization*, (1979), 325~338, Wiley, Chichester.
- Alfonso Misevicius, "Using Iterated tabu search for the traveling salesman problem", ISSN 1392-124X *INFOR MACINES TECHNOLOGIJOS IR VALDYMAS*, Vol.3, No.32(2004).
- Baker, B. M. and M. A. Ayechev, "A genetic algorithm for the vehicle routing problem", *Computer and Operational Research*, Vol.30 (2003), 787~800.
- Gillett, B. and L. Miller. "A heuristic algorithm for the vehicle dispatch problem", *Operational Research*, Vol.22(1974), 340~349.
- Laporte G., M. Gendreau, J-Y. Potvin, F. Semet, "Classical and modern heuristics for vehicle routing problem", *Ecole des HautesEtudes Commerciales and Centre for research on Transportation*, Montreal, (1999).
- Ursani, Z., R. Sarker, H. A. Abbass, "Improving the Performance of Genetic Algorithm in Capacitated Vehicle Routing Problem using Self Imposed Constraints" *Proceedings of the 2007 IEEE Symposium on Computational Intelligence in Scheduling(CI-Sched 2007)*.
- Gendreau, M., A. Hertz, and G. Laporte, "A tabu search heuristic for the vehicle routing problem", *Management Science*, Vol.40, No.10 (1994), 1276~1290.
- Osman, I. H., Netastrategy, "simulated annealing and tabu search algorithms for the vehicle routing problem", *Ann Operational Research*, Vol.41(1993), 421~451.
- Stefan, R., "Polynomial time heuristic for the VRP", *Topics in transportation and cargo loading*, september, (2005).

Abstract

## The Bisection Seed Detection Heuristic for Solving the Capacitated Vehicle Routing Problem

Jun-Taek Ko<sup>\*</sup> · Young-Hoon Yu<sup>\*</sup> · Geun-Sik Jo<sup>\*\*</sup>

The Capacitated Vehicle Routing Problem (CVRP) is the problem that the vehicles stationed at central depot are to be optimally routed to supply customers with demands, satisfying vehicle capacity constraints. The CVRP is the NP-hard as it is a natural generalization of the Traveling Salesman Problem (TSP).

In this article, we propose the heuristic algorithm, called the bisection seed detection method, to solve the CVRP.

The algorithm is composed of 3-phases. In the first phase, we work out the initial cluster using the improved sweep algorithm. In the next phase, we choose a seed node in each initial cluster by using the bisection seed detection method, and we compose the route with the nearest node from each seed. At this phase, we compute the regret value to decide the list of priorities for the node assignment. In the final phase, we improve the route result by using the tabu search and exchange algorithm.

We compared our heuristic with different heuristics such as the Clark-Wright heuristic and the genetic algorithm. The result of proposed heuristic show that our algorithm can get the nearest optimal value within the shortest execution time comparatively.

**Key Words** : Capacitated Vehicle Routing Problem, Bisection Seed Detection Heuristic, Sweep Algorithm, Tabu Search, Exchange Algorithm

---

\* School of Information Engineering, Inha University

\*\* School of Computer and Information Engineering, Inha University

## 저자 소개



고준택

인하대학교 컴퓨터정보공학 학사를 취득하고 동 대학원의 정보공학과 석사(2008)를 취득하였다. 주요 관심분야는 Resource Optimization, Intelligent Vehicle Routing, Constraint Programming, Intelligent Agent, Information Retrieval 등이다.



유영훈

관동대학교 전자계산공학과 학사, 인하대학교 전자계산공학 석사를 취득하였으며, 동 대학원의 정보공학과 박사과정을 수료하였다. 현재 인하대학교 컴퓨터정보공학부 강의전임교원으로 재직 중이다. 주요 연구분야는 인공지능, CSP, Constraint Programming, Intelligent Agent 등이다.



조근식

인하대학교 전자계산학과 학사, 미국 뉴욕 CUNY(City University of New York) 전자계산 석사와 전자계산 박사를 취득하였다. 한국지능정보시스템학회 편집장(1998), 인하대학교 창업지원센터 소장(2000), 인하대학교 전자계산소 소장(2005), 한국지능정보시스템학회 회장(2008) 등을 역임하였고, 현재 인하대학교 컴퓨터 정보공학부 교수로 재직하고 있으며 BK21 정보기술 사업 단장을 역임 중이다. 주요 연구분야는 인공지능, CSP, Intelligent E-Commerce System, Semantic web 등이며, AI Magazine, Expert System with Application, Journal of Organizational Computing and Electronic Commerce 등 다수의 학술지에 논문을 게재하였으며, 다수의 특허를 보유하고 있다.