

식품 배송의 특성을 고려한 차량경로문제의 발견적 해법

강경환 · 이영훈[†]

연세대학교 정보산업공학과

Heuristic for Vehicle Routing Problem with Perishable Product Delivery

Kyung Hwan Kang · Young Hoon Lee

Department of Information and Industrial Engineering, Yonsei University, Seoul 120-749

The purpose of Vehicle Routing Problem (VRP) is to design the least costly (distance, time) routes for a fleet of identically capacitated vehicles to serve geographically scattered customers. There may be some restrictions such as the maximal capacity for each vehicle, maximal distance for each vehicle, time window to visit the specific customers, and so forth. This paper is concerned with VRP to minimize the sum of elapsed time from departure, where the elapsed time is defined as the time taken in a moving vehicle from the depot to each customer. It is important to control the time taken from departure in the delivery of perishable products or foods, whose freshness may deteriorate during the delivery time. An integer linear programming formulation is suggested and a heuristic for practical use is constructed. The heuristic is based on the set partitioning problem whose performances are compared with those of ILOG dispatcher. It is shown that the suggested heuristic gave good solutions within a short computation time by computational experiments.

Keywords: Vehicle Routing Problem, Set Partitioning Problem

1. 서론

차량경로문제는 창고에서 출발한 차량이 지리적으로 산재된 고객을 서비스하기 위한 최적의 차량경로를 결정하는 문제로서 최적해를 구하기 위한 계산시간이 지수적으로 요구되는 조합최적화(Combinatorial optimization) 문제이다. 차량경로문제의 목적함수는 차량의 총 이동시간(거리)을 최소화하는 것이며, 창고의 가용차량의 수, 차량의 적재용량, 고객수요, 차량의 이동시간 제한 등의 제약조건이 주어진다.

차량경로문제에 대한 연구는 물류산업의 현실적 제약조건과 목적함수를 반영하기 위한 모델개발의 연구와 합리적 시간 내에 근사해를 제공하는 발견적해법 개발의 연구형태로 진행되어왔다. 대표적인 차량경로문제의 응용모델은 서비스 시간 제약이 있는 차량경로문제(VRPTW : VRP with Time Window)이다. 이 문제는 고객이 요구하는 서비스 시간의 상하한이 준

재하며, 이 서비스 시간대를 위반하였을 경우 해가 성립이 되지 않는 Hard Time Window 문제와 실행가능한 해로 정의하되 벌금비용이 부과되는 Soft Time Window 문제로 나뉘어진다. Hard Time Window 문제는 Rodolfo and Jaime(2006), David and Olli(2005), Olli *et al.*(2004) 등에 의해 연구되었으며, Soft Time Window 문제는 Hideki *et al.*(2006), Herminia *et al.*(2005), George *et al.*(2003) 등에 의해 다양한 알고리즘이 보고되었다.

다회방문 차량경로문제(MTVRP: Multi-Trip VRP)는 각 차량의 주어진 이동시간 제한의 제약조건내에서 창고를 여러 번 출발할 수 있는 문제이다. 물류산업현장에서의 차량운행계획은 대부분 다회방문 형태로 수립되어 있으며, 차량의 적재용량과 가용차량수의 제한, 고객 서비스 시간대 제약 등의 조건으로 인해 발생하게 된다. 이 문제에 대해 Fagerholt(1999)는 1 단계에서 모든 실행가능한 해를 생성하고 2 단계에서 최적의 해를 선정하는 2단계 발견적해법을 제시하였으며, Alfredo and

[†] 연락저자 : 이영훈, 120-749 서울시 서대문구 신촌동 134, 연세대학교 정보산업공학과, Tel : 02-2123-4813, Fax : 02-364-7807
E-mail : youngh@yonsei.ac.kr

Omar(2005)는 Adaptive memory 알고리즘을 적용하였다.

이러한 차량경로문제(HVRP: Heterogeneous VRP, VFM: Vehicle Fleet Mix)는 창고에서 운용가능한 차량의 적재용량이 상이한 문제로서, 운행되는 차량의 고정비용과 이동시간에 따른 변동비용이 함께 고려되는 VFMFC(VFM with fixed cost) 문제와 변동비용만 고려되는 VFMVC(VFM with variable cost)로 나눌 수 있다. Lee *et al.*(2006)은 타부서치간 생성된 이웃해 중에서 최적의 해를 선정하기 위해 집합분할 문제(Set Partitioning)를 적용하였다. Salhi and Rand(1993)는 현재해에서 다양한 이웃해를 탐색하기 위해 축소(reduction), 재할당(reassignment), 조합(combination)등의 해법을 제시하였으며, 유전알고리즘, 타부서치에 의한 메타휴리스틱 측면에서의 연구는 Ochi(1998)와 Gendreau *et al.*(1999)등에 의해 보고되었다.

또 다른 형태의 차량경로문제의 응용형태는 배달과 수거가 혼합된 형태의 문제이다(PDPVRP: Pick-up and Delivery VRP). 이 문제는 고객이 요구하는 서비스의 종류가 고객에 따라 상이한 문제로서, 어떤 고객은 배송을 원하고, 어떤 고객은 수거를 요구하는 형태이다. Mosheiov(1998)는 현재의 경로를 다른 차량에 의해 서비스를 받을 수 있는 여러개의 경로로 분할하여 해를 탐색하는 tour partitioning 해법을 제시하였다. 이 외에도 시간상한을 가지는 차량경로문제(TDVRP: Time Deadline VRP), 복수창고 차량경로문제(MDVRP: Multi-Depot VRP), 한 고객이 여러대의 차량으로부터 서비스를 받을 수 있는 분할배송 차량경로문제(SDVRP: Split Delivery VRP), 이동시간, 고객의 수요량, 고객의 수 등이 확정적이지 않은 차량경로문제(SVRP: Stochastic VRP)등이 차량경로문제의 응용모델로 연구되어왔다.

차량경로문제의 복잡성(Complexity)으로 인해 발견적해법 개발에 많은 연구들이 보고되었다. 발견적해법의 접근은 최적화 알고리즘(Exact algorithm), 전통적 발견적 해법(Classical heuristic), 메타휴리스틱 등으로 구분된다(Kang, 2006). 최적화 알고리즘은 최적해를 제공한다는 장점은 있으나 계산시간은 비교적 길다. 최적화 알고리즘은 K-tree 방법(Fisher, 1994), 분지한계법(Blasum and Hochstättler, 2000; Ladányi *et al.*, 2001; Lee and Mitchell, 1998; Toth and Vigo, 2001), 동적계획법(Lee *et al.*, 2006; Malandraki and Dial, 1996) 등으로 연구되었다. 전통적 발견적 해법은 주어진 평가함수를 통해 해를 구축해나가는 construction 알고리즘과 해를 개선하기 위해 교환 삽입, 크로스 등의 반복적인 절차를 수행해 나가는 improvement 알고리즘으로 구분할 수 있다. Construction 알고리즘은 절약방법(Saving method Clark and Wright, 1864), 스위핑방법(Sweeping method Gillet and Miller, 1974), 근접이웃해 방법(Nearest to neighbor method Rosenkrantz *et al.*, 1974)등의 해법이 제시되었다. Improvement 알고리즘은 경로내 해의 개선방법인 intra-route 개선방법(2-opt, Or-opt)과 경로의 해의 개선방법(cross, exchange, relocate) 등이 연구되었다(ILOG, 2002). 메타휴리스틱으로 유전알고리즘, 타부서치, 시뮬레이티드 어닐링등의 방

법이 수행되었다(Kim, 2006; Tavakkoli-Moghaddam *et al.*, 2006a; Tavakkoli-Moghaddam *et al.*, 2006b). Soft time-window 차량경로문제와 같은 형태에서는 벌금비용의 형태로서 고객의 서비스 만족에 대한 목적을 가진 연구도 있었지만 대부분의 차량경로문제의 응용모델은 서비스 제공자의 관점에서 개발되었다. 그에 따라, 목적함수는 차량의 총 이동거리의 최소화 총 이동시간의 최소화, 차량의 고정비용의 최소화 등으로 이루어져 있다. 현재의 수, 배송 환경에서 고객의 서비스 만족도는 의사결정시에 반드시 고려해야 할 요소중의 하나이다.

본 연구는 고객의 측면에서 모델링된 새로운 형태의 차량경로문제를 제시한다. 본 연구의 목적함수는 제품이 차량에 적재된 이후 고객에게 배송되기까지 경과된 시간의 합(sum of the elapsed time)으로 정의된다. 이 문제가 가지는 의미는 다음과 같다. 신선함을 요하는 제품(야채, 생선, 빵 등)은 가능한 빠른 시간내에 고객에게 배송하는 것이 중요하다. 배송차량에 적재되어 있는 시간이 길면 길수록 이러한 제품의 가치는 감소하기 때문이다. 즉, 시간의 경과에 따라 그 가치가 감소하는 제품들에 적합한 모델로서 경과된 시간이란 손실된 그 제품의 가치로 볼 수 있다. 또한 제품이 배송된 이후, 최종적인 고객에게 그 제품을 판매하기 위해 준비작업이 필요한 제품의 경우에도 적합한 모델이다. 예를 들어 신문배송의 경우, 중앙지사에서 지역지사를 거쳐 최종 구독자에게 배송업무가 이루어진다. 각 지역지사의 경우 중앙지사로부터 가능한 빨리 신문을 배송을 받기를 위한다. 왜냐하면, 최종 구독자에게 배송하기 위해서는 각 지역별 신문분류, 광고지 삽입 등의 추가적인 업무 소요가 있기 때문이다. 이러한 측면에서, 본 연구가 제시하는 목적함수는 생산스케줄링 문제의 flow-time의 최소화와 같다.

발견적 해법은 집합분할(Set Partitioning)의 개념에 기반을 둔 것으로, 2단계의 알고리즘으로 구성된다. 1단계에서 기존의 construction 알고리즘(sweeping, nearest to neighbor method)을 응용하여 다양한 경로를 생성하고, 2단계에서 집합분할의 최적화 수리모델을 이용하여 최적의 경로를 선정하게 된다. 제시된 알고리즘의 평가는 ILOG dispatcher의 알고리즘과 비교되었다. 알고리즘이 가지는 기여점은 첫째는 다양한 경로의 생성과정이다. 기존의 construction 알고리즘만으로는 제시된 모델에 대한 좋은 해를 제공할 수 없으며, 단일의 해를 제공하므로 목적함수에 적합한 다양한 경로생성은 불가능하다. 이에 따라, 스위핑 방법과 근접 이웃해 방법에 기반을 둔 경로생성 메커니즘을 제시하였고, 이를 통해 2단계 해법을 위한 다수개의 좋은 경로를 생성하였다. 둘째는 이와 같은 다수개의 경로 중에서 최적의 조합을 찾는 문제를 집합분할(Set partitioning) 개념을 적용하였다는 것이다. 집합분할문제의 최적화 수리모델만으로는 차량경로문제에 대한 답을 제공할 수 없다. 차량경로문제를 풀기위한 해법으로서 집합분할문제를 적용하기 위해 원문제를 변화해나가는 과정(경로생성, 경로최적화)에서 알고리즘의 기여가 있다. 모든 가능한 경로를 생성하여 집합분할의 수리모델을 이용할 경우 최적해는 보장되지만 합리적

시간내에 해는 제공해주지 못한다. 본 2단계 알고리즘은 생성되는 경로의 수는 줄어디, 좋은 경로를 생성함으로써 짧은 시간에 좋은 해를 선정해주게 된다. 집합분할 최적화 수리모델은 기존의 차량경로문제가 가지는 계산복잡성을 상당히 줄이는 효과를 가져다 준다. 제 2장에서는 본 논문이 제시하는 차량경로문제의 최적화 수리모델이 제시되고 제 3장에서는 발견적 해법, 제 4장과 제 5장에서는 실험결과와 결론을 제시한다.

2. 최적화 수리모델

최적화 수리모델은 하나의 창고에서 가용한 K 개의 차량을 활용하여 N 개의 고객에게 제품을 배송하는 환경에서 구현된다. 각 차량은 동일한 적재용량 U 를 가지고 있으며, 각 고객은 수요량은 사전에 알려져 있다. 각 지점간 이동시간은 확정적이고, 각 차량의 이동시간에 대한 제한은 없으며 모든 고객들에 대해 배송이 이루어져야 한다. 각 차량은 창고에서 출발하여 창고로 복귀한다. <Figure 1>은 실행가능한 차량운행계획의 예제이다.

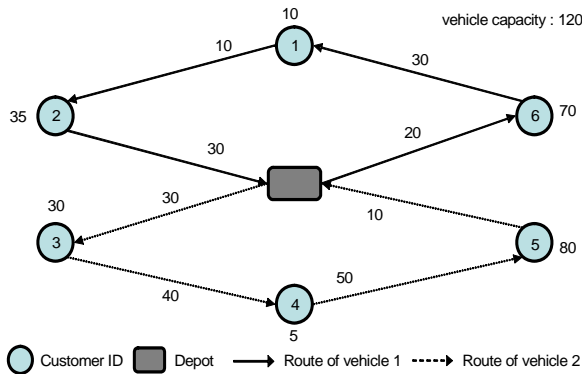


Figure 1. The example of VRP with perishable product delivery

차량 1은 70, 10, 35의 수요량을 고객 6, 1, 2의 순서로 배송하며, 차량 2는 30, 5, 80의 고객 수요량을 고객 3, 4, 5의 순서대로 배송을 한다. 각 차량 모두 적재용량 120을 초과하지 않으며, 창고에서 출발한 차량은 창고로 복귀한다 이때 고객 6이 요구하는 제품의 경우 차량에 적재되어 배송되기까지는 20이라는 시간이 경과한다. 고객 1의 경우는 고객 6을 거쳐서 배송이 이루어지므로 창고에서 고객 6까지의 이동시간과 고객 6에서 고객 1까지의 이동시간의 합인 50의 시간이 경과한 후에 배송을 받게된다. <Table 1>은 총 목적함수를 나타낸 것이다. 전통적인 차량경로문제의 목적함수가 각 차량의 도착시간인 90과 130의 합인 220인 반면에 제시되는 모델의 목적함수는 각 고객이 요구하는 제품이 출발로부터 경과된 시간의 합인 570이 된다.

Table 1. The objective function of VRP regarding perishable product delivery

Vehicle	Service sequence	Arrival times	Sum of elapsed time
Vehicle 1	6	20	20
	1	50	20 + 30 = 50
	2	60	20 + 30 + 10 = 60
	Depot	90	20 + 30 + 10 + 30 = 90
Vehicle 2	3	30	30
	4	70	30 + 40 = 70
	5	120	30 + 40 + 50 = 120
	Depot	130	30 + 40 + 50 + 10 = 130
Ob.fn. of standard VRP	90 + 130 = 220		
Ob. fn. of suggested model	(20 + 50 + 60 + 90) + (30 + 70 + 120 + 130) = 570		

<기호>

i, j : 고객 인덱스 ($i, j = 1, 2, \dots, N$, 1 은 창고를 의미함)

k : 차량 인덱스 ($k = 1, 2, \dots, K$)

s : 서비스 순서 인덱스 ($s = 1, 2, \dots, S, S = N$, 예를 들어 각 차량의 경로에서 마지막으로 서비스를 받는 고객과 창고까지의 트립은 " $s = 1$ "로 정의됨.)

U : 차량의 적재용량

D_i : 고객 i 의 수요량

d_{ij} : 고객 i 와 고객 j 간의 이동시간

T : 차량에 의해 서비스 받는 고객의 부분집합

V : 차량에 의해 서비스 받는 고객의 전체집합($V \setminus \{1\}$: 창고는 제외함).

$X_{i,j,k,s}$: 만약 차량 k 가 고객 i 와 j 사이를 마지막에서 s 번째 이동이 있으면 1 그렇지 않으면 0.

<최적화 수리모델>

$$\text{Min} \sum_i \sum_j \sum_k \sum_s s d_{i,j} X_{i,j,k,s} \quad (2.1)$$

subject to

$$\sum_i \sum_k \sum_s X_{i,j,k,s} = 1 \quad \forall j \quad (2.2)$$

$$\sum_j \sum_k \sum_s X_{i,j,k,s} = 1 \quad \forall i \quad (2.3)$$

$$\sum_i \sum_j X_{i,j,k,s} = 1 \quad \forall k, s \quad (2.4)$$

$$\sum_i X_{i,p,k,s} - \sum_j X_{p,j,k,s+1} = 0 \quad \forall k, s, p = 2, \dots, N \quad (2.5)$$

$$\sum_i D_i \sum_j \sum_s X_{i,j,k,s} \leq U \quad \forall k \quad (2.6)$$

$$\sum_{i \in T} \sum_j \sum_s X_{i,j,k,s} \leq |T-1| \quad \forall k, T \subseteq V, T \neq \emptyset \quad (2.7)$$

$$X_{i,i,k,s} = 0 \quad \forall i, k, s \quad (2.8)$$

$$X_{i,j,k,s} = 0 \text{ or } 1 \quad \forall i, j, k, s \quad (2.9)$$

제시하는 모델의 목적함수는 식 (2.1)과 같이 서비스 순서 인덱스인 s 를 포함시키고 결정변수를 s 가 포함된 4차원의 형태로 표현함으로써 목적함수 도출이 가능하다. 예를 들어 <Table 1>과 같이 창고에서 첫 번째 서비스 지점인 6번 고객까지의 이동시간인 20이 목적함수에 4번에 걸쳐 누적되며 이때 $s = 4$ 가 된다. 창고가 고객으로서 간주될 경우 식(2.1)의 목적함수를 이용하고, 순수한 고객만을 대상으로 한 목적함수는 다음과 같다.

$$Min \sum_i \sum_j \sum_k \sum_s (s-1)d_{i,j} X_{i,j,k,s} \quad (2.10)$$

각 고객이 차량에 의해 한번만 서비스될 수 있음을 나타내는 식이 (2.2)와 (2.3)이다. 식 (2.4)는 모든 고객은 각 차량의 서비스 순서에 한번만 할당됨을 의미한다. 식 (2.5)는 차량 흐름의 연속성을 의미한다. 특정 지점 p 에 도착한 차량은 그 지점에서 출발해야하며, 서비스 순서인덱스인 s 는 1이 증가한다. 식 (2.6)는 차량 적재용량의 제약조건이다. 식 (2.7)은 부분경로 방지(sub-tour elimination) 제약조건으로 창고를 포함하지 않는 경로형성을 방지한다. 식 (2.8)은 동일지점간 이동은 금지하게 만드는 제약조건이다. 제시된 최적화 수리모델은 문제의 크기가 커질수록 최적해를 구하기 위한 계산시간이 지수적으로 증가한다.

3. 집합분할 발견적해법(SPH : Set Partitioning based Heuristic)

집합분할은 2단계의 해법으로 구성된다. 1단계에서는 스위핑 방법(Gillett and Miller, 1974)과 근접이웃해 방법(Rosenkrantz, 1974)을 응용하여 목적함수에 적합한 다양한 경로들을 생성한다. 2단계에서는 생성된 다양한 경로중에서 원문제의 제약조건에 맞는 경로를 선정하며, 이 과정은 집합분할문제의 최적화 수리모델을 통해 풀리게 된다.

3.1 경로 생성(Route generation)

각 차량으로부터 서비스를 받는 고객의 집합(클러스터링)은 스위핑 방법에 의해 생성한다. 스위핑 방법은 씨앗고객(seed node)의 선정과 스위핑 방향시계, 반시계 방향에 따라 고객의 집합이 달라진다. 창고를 포함한 고객이 N 개일 경우, 총 $2(N-1)$ 개의 클러스터링이 생성된다. 각 클러스터링 내에서 2 단계해법을 위한 경로를 생성하는 과정은 근접 이웃해 개념을 적용하였다. 근접 이웃해 개념을 적용한 이유는 목적함수 값이 가지는 특성 때문으로, 앞서 방문하는 고객까지의 이동시

간이 목적함수에 누적되기 때문이다. 식 (3.1)과 같이 어떤 차량에 의해 서비스를 받는 고객의 수가 m 일 경우, 창고로부터 첫 번째로 서비스를 받는 고객까지의 이동시간 $t_{(1)}$ 은 목적함수에 m 배 누적된다. 첫 번째로 서비스를 받는 고객과 두 번째로 서비스를 받는 고객까지의 이동시간은 $(m-1)$ 배 누적된다. 이와 같은 특성으로 인해 짧은 이동시간을 가지는 고객이 앞서 서비스를 받는 것이 바람직하다.

$$\begin{aligned} \text{Objective function} &= m \cdot t_{(1)} + (m-1) \cdot t_{(2)} + \\ &\dots + 2 \cdot t_{(m-1)} + t_{(m)} \end{aligned} \quad (3.1)$$

각 클러스터링 내에서 경로를 생성하는 과정은 <Figure 2>와 같다.

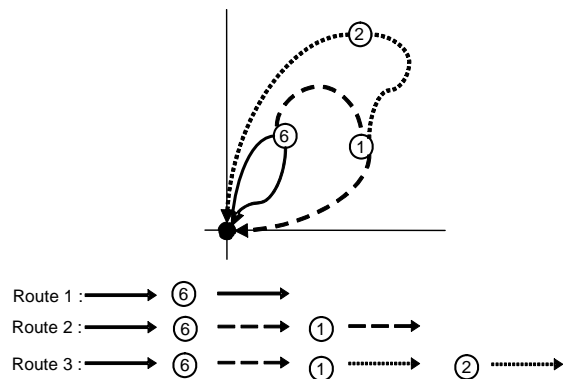


Figure 2. The example of route generation

예를들어 클러스터링이 고객 1, 6, 2가 되었다고 가정하면, 생성되는 경로의 수는 3개로서, 첫 번째는 창고로부터 가장 가까운 고객 6번에 대해서만 배송하고 복귀하는 경로이며, 두 번째 경로는 6번 고객에 대한 배송후에 6번 고객과 가장 가까운 1번 고객에 대한 서비스를 제공후 복귀하는 경로이다. 마지막은 고객 6, 1, 2번 고객을 서비스하고 복귀하는 경로이다. 앞서 생성된 $2(N-1)$ 개의 클러스터링에 대해 세부 경로를 생성하면, 총 $2(N-1)^2$ 개의 세부 경로가 생성되며, 이 중에서 최적의 경로의 조합은 2단계의 경로 최적화 과정을 통해 선정된다.

3.2 경로 최적화(Route optimization)

2단계 경로 최적화단계는 1단계에서 생성된 총 $2(N-1)^2$ 개의 세부 경로를 대상으로 각 고객이 한번씩만 포함이 되는 최적 경로를 찾는 과정이다. 예를 들어 <Figure 2>에서의 6번 고객은 경로 1, 2, 3에 모두 포함되어 있으며, seed node의 변화에 따라, 또 다른 clustering에서 생성된 경로상에도 6번 고객은 포함될수 있다. 최종적으로 feasible한 차량경로가 생성되기 위해서는 6번고객이 포함된 단 한 개의 경로만 선정이 되어야 하며 최종적으로 선정되는 경로의 수는 운행되는 차량의 수와 동일하므로, 가용 차량수 K 개를 초과할 수 없다. 목적함수 (3.2)는 경로비용의 합을 최소화 하는 것으로 원문제의 목적함수와 동

일한 의미를 가진다. 식 (3.3)은 각 고객이 한번만 포함되도록 하며, 식 (3.4)는 가용 차량수 제약조건이며 식(3.5)는 결정변수의 이진변수 조건이다.

<기호>

r = 경로 인덱스 ($r = 1, \dots, 2(N-1)^2$)

c_r = 1단계에서 생성된 r 경로의 비용

K = 가용 차량의 수

y_r = 만약 경로 r 이 선정되면 1, 그렇지 않으면 0

F_i = 고객 i 를 포함하는 경로

<최적화 수리모델>

$$\text{Min} \sum_r c_r y_r \tag{3.2}$$

subject to

$$\sum_{r \in F_i} y_r = 1 \quad \forall i \tag{3.3}$$

$$\sum_r y_r \leq K \tag{3.4}$$

$$y_r = 0 \text{ or } 1 \quad \forall r \tag{3.5}$$

Table 2. Experimental result of small scale instances

Set	Instance	Optimal solution		SPH		ILOG dispatcher		Gap(%)	
		Ob. fn.	C.T.(sec.)	Ob. fn.	C.T.(sec.)	Ob. fn.	C.T.(sec.)	Opt. vs. SPH	Opt. vs. ILOG dispatcher
20 customers/ 2 vehicles	1	400	2033	403	0.02	403	88	0.72	0.72
	2	238	1609	242	0.17	245	75	1.75	2.86
	3	237	631	245	0.01	237	84	2.92	0.00
	4	261	1086	267	0.02	261	84	2.08	0.00
	5	288	9932	300	0.01	288	82	4.27	0.00
	6	267	2584	273	0.02	267	81	2.17	0.00
	7	496	1596	496	0.02	496	75	0.00	0.00
	8	455	1652	479	0.02	455	65	5.03	0.00
	9	453	402	459	0.01	453	84	1.28	0.00
20 customers/ 3 vehicles	1	271	1689	278	0.00	271	86	2.61	0.00
	2	254	4003	260	0.01	254	89	2.33	0.00
	3	237	4084	237	0.02	237	78	0.10	0.00
	4	235	4837	242	0.01	239	85	2.93	1.55
	5	240	2987	243	0.02	240	78	1.42	0.00
	6	243	1270	251	0.00	243	82	3.40	0.00
	7	339	1452	349	0.02	344	86	3.02	1.57
	8	368	864	379	0.01	368	84	2.84	0.00
	9	346	135	377	0.00	363	87	8.30	4.63
20 customers/ 4 vehicles	1	214	125	220	0.01	214	98	2.72	0.00
	2	209	16668	213	0.01	209	92	1.69	0.00
	3	228	5421	243	0.02	228	106	6.18	0.00
	4	250	3785	253	0.00	250	98	1.22	0.00
	5	203	1349	203	0.02	203	95	0.00	0.00
	6	221	1254	230	0.01	221	102	3.89	0.00
	7	299	15	299	0.02	303	102	0.00	1.38
	8	283	3562	284	0.01	286	87	0.27	0.78
	9	313	521	339	0.01	313	103	7.59	0.00
20 customers/ 5 vehicles	1	254	2875	256	0.01	257	98	0.84	1.08
	2	264	2658	274	0.02	267	94	3.82	1.43
	3	264	1452	287	0.02	297	86	8.15	11.15
	4	219	13405	219	0.01	219	87	0.00	0.00
	5	210	10130	224	0.03	210	75	6.18	0.00
	6	229	7852	232	0.01	229	94	1.25	0.00
	7	323	56	332	0.01	323	119	2.62	0.00
	8	292	1542	308	0.01	292	75	5.14	0.00
	9	392	3521	392	0.02	397	96	0.06	1.28
Average		286	3306	294	0.02	288	88	2.68	0.79

Ob.fn. : Objective function, C.T. : Computation time

4. 실험결과

집합분할 발견적해법의 성능은 ILOG dispatcher의 성능과 비교하였다. ILOG dispatcher 알고리즘은 다음과 같이 구현하였다. 초기해는 절약해법(Saving method)을 이용하여 구하고, 해의 개선과정은 경로내 개선방법인2-opt, Or-opt와 경로외 개선방법인 cross, exchange, relocate 방법을 적용하였다. 이러한 알고리즘은 Lin(1965), Lin and Kernighan(1973), Timon *et al.* (2005) 등에 의해 보고되었다.

실험방법은 두 가지 형태로 진행하였다. 첫째는 최적해와의 비교로서 최적해를 구하는 계산시간의 한계로 고객수는 20개로 고정하고, 차량수를 2대에서 5대까지 증가시켰으며, 실험 예제의 수는 총 36개를 생성하였다. 두 번째는 집합분할 해법과 ILOG dispatcher의 비교로서, 사용된 데이터는 Solomon (1988) 데이터 18개와 Breedam(1994, <http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html>) 데이터(T1 데이터 60개)를 사용하였다. 두 가지의 벤치마크 예제에서, 본 모델과 관련이 없는 서비스 하한, 상한, 서비스 시간 등은 제외하였으며, 고객과 창고의 좌표, 수요량, 가용차량 수, 차량의 적재용량은 기존 데이터와 동일하다. Breedam(1994) 데이터의 경우 가용차량수가 주어지지 않기 때문에 Solomon(1988) 데이터와 동일한 25대의 가용차량수를 적용하였다. 최적해는 상용 tool인 ILOG사의 CPLEX 8.1을 이용하여 구했으며, 사용된 컴퓨터는 펜티엄 4 CPU 2.6 GHz 이다.

<Table 2>은 최적해와의 성능비교 결과이다. 최적해를 구하

기 위한 계산시간은 36개의 예제에 대해 평균 3,306초 소요되었으며, 20개 이상의 고객이 있는 예제에 대해서는 최적해를 구하기가 어렵다. 집합분할 발견적해법은 1초 이내에, ILOG dispatcher는 평균 88초 소요되었다. Gap은 $(SPH \text{ value} - \text{optimal solution}) \times 100 / SPH \text{ value}$, $(ILOG \text{ dispatcher value} - \text{optimal solution}) \times 100 / ILOG \text{ dispatcher value}$ 로 정의하였다. 집합분할 해법은 최적해 대비 평균 2.68%의 gap을 보였으며, ILOG dispatcher는 최적해 대비 0.79%의 gap을 유지하였다.

벤치마크 예제에 대한 실험결과는 <Table 3>과 <Table 4>와 같다. Gap은 $(SPH \text{ value} - ILOG \text{ dispatcher value}) \times 100 / ILOG \text{ dispatcher value}$ 로 정의하였다. Solomon(1988) 예제의 경우 목적함수 측면에서는 ILOG dispatcher 알고리즘 보다 집합분할 해법이 1.81% 높으나, 계산시간 측면에서는 집합분할 해법이 평균 0.08초 소요된 반면, ILOG dispatcher 알고리즘은 약 980초 소요되었다. Breedam(1994) 데이터의 경우 각 set은 10개의 instance로 구성되어 있으며 <Table 4>에 제시된 값은 각 set의 평균치를 나타낸다. 목적함수 측면에서는 ILOG dispatcher 알고리즘에 비해 집합분할 해법의 값이 0.76% 높게 나왔으나, 계산시간 측면에서는 상당한 절감이 이루어졌다. ILOG dispatcher의 경우, 해의 개선과정간 적용되는 inter-route/intra-route 알고리즘의 특성인 랜덤성으로 인해 문제의 규모가 커짐에 따라 해의 탐색영역이 커지게되며, 이는 계산시간의 증가원인이 된다. 집합분할 해법과 ILOG dispatcher와의 gap은 평균 1.81%와 0.76%로서, 계산시간에 비해 상당히 우수한 해를 집합분할 해법이 제공함을 알 수 있다. 이는 다양하고 우수한 경로생성이라는

Table3. Experimental result of Solomon(1988)'s instances

Instance	Number of customers	Number of vehicles	Vehicle capacity	C.T(second)		Objective Function		Gap(%)
				SPH	ILOG dispatcher	SPH	ILOG dispatcher	
C101	25	25	200	0.05	28	914	953	-4.09
C101	50	25	200	0.05	201	1909	1869	2.14
C101	100	25	200	0.11	1451	4433	4436	-0.07
C201	25	25	700	0.02	45	1095	1071	2.16
C201	50	25	700	0.16	360	2293	2192	4.61
C201	100	25	700	0.09	2680	4800	4636	3.54
R101	25	25	200	0.01	36	1304	1291	1.08
R101	50	25	200	0.05	289	2472	2436	1.48
R101	100	25	200	0.11	1683	4201	4135	1.59
R201	25	25	1000	0.01	58	1304	1297	0.59
R201	50	25	1000	0.02	602	2472	2387	3.54
R201	100	25	1000	0.11	4773	4201	4136	1.58
RC101	25	25	200	0.01	13	1469	1450	1.32
RC101	50	25	200	0.03	44	2982	2964	0.61
RC101	100	25	200	0.03	1575	5372	5158	4.15
RC201	25	25	1000	0.13	38	1470	1450	1.41
RC201	50	25	1000	0.17	269	3028	2975	1.77
RC201	100	25	1000	0.23	3503	5372	5106	5.21
Average				0.08	980	2838	2774	1.81

C.T. : Computation time

Table 4. Experimental result of Breedam(1994)'s instances

Set	Instance	Number of customers	Number of vehicles	Vehicle capacity	C.T(second)		Objective Function		Gap(%)
					SPH	ILOG dispatcher	SPH	ILOG dispatcher	
1	1T1~10T1	100	25	100	0.19	978	7474	7343	1.78
2	11T1~20T1	100	25	100	0.46	1161	7022	6945	1.11
3	21T1~30T1	100	25	100	0.27	756	7034	7009	0.36
4	31T1~40T1	100	25	100	0.21	1274	5483	5477	0.11
5	41T1~50T1	100	25	100	0.18	849	5948	5929	0.33
6	51T1~60T1	100	25	100	0.25	1113	6512	6455	0.89
Average					0.26	1022	6579	6526	0.76

C.T. : Computation time

1단계와 그 중에서 최적의 경로를 선정하는 2단계해법이 효과적임을 보여준다.

6. 결론 및 향후 연구방향

기존의 차량경로문제의 응용모델이 주로 공급자의 관점에서 모델링된 반면, 본 연구에서 제시하는 차량경로문제는 고객의 관점에서 모델링되었다. 즉, 수송 차량에 적재되어 있는 시간 일 길어질수록 그 제품의 가치가 감소하는 신선함을 요하는 식품이나, 각 고객의 입장에서 빠른 배송을 원하는 수송업무에서 적용할 수 있는 차량경로문제이다. 이와 같은 차량경로문제의 목적함수는 제품이 차량에 적재된 이후 고객에게 배송되기까지 경과된 시간의 합(sum of the elapsed time)으로 정의된다. 발견적 해법으로 집합분할 발견적 해법을 제시하였다. 1 단계에서 차량의 경로를 다수개를 생성하였고 2단계에서 집합분할의 최적화 수리모델을 통해 최적의 경로를 선정한다. 계산 복잡성이 큰 차량경로문제의 해법을 복잡성이 낮은 집합분할의 개념을 적용하기 위해 원문제를 효과적으로 변화시켜가는 1, 2단계의 접근이 효과적임이 실험결과 입증되었다.

향후 연구는 응용모델의 연구와 발견적 해법의 개선이라는 측면에서 추진될 수 있다. 각 고객의 중요도가 상이한 형태 분할배달이나 복수창고를 가지는 환경 차량운행계획 기간이 다 기간으로 확장된 상황하에서의 모델개발도 의미가 있을 것이다. 제시한 집합분할의 해법은 해의 탐색공간이 제한되어 있다는 단점이 있다. 즉 1단계에서 생성되는 경로의 수가 $2(N-1)^2$ 으로 계산시간은 짧으나 해의 성능측면에서는 개선의 여지가 있다고 판단된다. 해의 탐색공간의 확대와 관련된 연구, 메타휴리스틱을 이용한 해법의 연구도 추진될 것이다.

참고문헌

Alfredo O. and Omar V. (2005), Adaptive memory programming

for the vehicle routing problem with multiple trips, *Computers and Operations Research*, **34**(1), 28-47.

Blasum, U. and Hochstättler, W. (2000), Application of the branch and cut method to the vehicle routing problem. Technical Report zaik2000-386, *Centre of Applied Computer Science*, University of Cologne, Germany.

Breedam, A. (1994), An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints. *Ph.D. dissertation, University of Antwerp*. 1994.

Clarke, G. and Wright J. (1964), Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, **12**(4), 568-581.

David M. and Olli B. (2005), Active-guided evolution strategies for large-scale capacitated vehicle routing problems, *Computers and Operations Research*, In Press, Corrected Proof, Available online.

Fagerholt, K. (1999), Optimal fleet design in a ship routing problem. *International Transactions In Operational Research*, **6**(5), 453-464.

Fisher, M. L. (1994), Optimal solution of vehicle routing problems using minimum K-trees. *Operations Research*, **42**, 626-642.

Gendreau, M., Laporte, G., Musaraganyi, C., and Taillard, E. D. (1999), A tabu search heuristics for the heterogeneous fleet vehicle routing problem. *Computers and Operations Research*, **26**(12), 1153-1173.

George I., Manolis, K., and Gregory, P. (2003), A problem generator-solver heuristic for vehicle routing with soft time windows, *Omega*, **31**(1), 41-53.

Gillett, B and Miller, L. (1974), A heuristic algorithm for the vehicle dispatching problem. *Operations Research*, **22**, 340-349.

Herminia I. C., Carmen, G., María-José O. and Belén, S. (2005), A goal programming approach to vehicle routing problems with soft time windows, *European Journal of Operational Research*, **177**(3), 1720-1733.

Hideki H., Toshihide I., Shinji I., and Mutsunori Y. (2006), The vehicle routing problem with flexible time windows and traveling times, *Discrete Applied Mathematics*, **154**(16), 2271-2290.

ILOG. (2002), ILOG Dispatcher 3.3 user's manual. *ILOG*, 49-92.

Kang, K. H. (2006), Vehicle routing/scheduling models considering customer satisfaction and vehicle availability, *Ph.D. dis-*

- sertation, *The graduate school of Yonsei university*.
- Kim, J. (2005), Heterogeneous vehicle routing problem with tabu search. *Ph.D. dissertation, The graduate school of Yonsei university*.
- Ladányi, L., Ralphs, T. K., and Trotter Jr., L. E. (2001), Branch, cut, and price : sequential and parallel, in *Computational Combinatorial Optimization*. D. Naddef and M. Juenger, eds., Springer, Berlin.
- Lee, C. G., Epelman, M., White III, C. C., and Bozer, Y. A. (2006), A shortest path approach to the multiple-vehicle routing problem with split picks-up. *Transportation Research – Part B : Methodological*, **40**(4), 265-284.
- Lee, E. K. and Mitchell, J. E. (1998), Branch-and-bound methods for integer programming. in the *Encyclopedia of Optimization*, Kluwer Academic Publishers.
- Lee, Y. H., Kim, J. I., Kang, K. H., and Kim, K. H. (2006), A heuristic for vehicle fleet mix problem using tabu search and set partitioning. *Journal of the Operational Research Society*, In the 3rd review.
- Lin, S. (1965), Computer solutions on the traveling salesman problem. *Bell Systems Technical Journal*, **44**, 2245-2269.
- Lin, S. and Kernighan, B. (1973), An efficient heuristic for the traveling salesman problem. *Operations Research*, **21**(2), 498-516.
- Malandraki, C. and Dial, R. B. (1996), A restricted dynamic programming heuristic algorithm for the traveling salesman problem. *European Journal of Operational Research*, **90**(1), 45-55.
- Mosheiov, G. (1998), Vehicle Routing with Pick-up and Delivery : Tour-partitioning heuristics. *Computers and Industrial Engineering*, **34**(3), 669-684.
- Ochi, L. S. (1998), A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. *Future Generation Computer Systems*, **14**(5-6), 285-292.
- Olli, B., Geir, H., and Wout, D. (2004), A multi-start local search algorithm for the vehicle routing problem with time windows. *European Journal of Operational Research*, **159**(3), 586-605
- Rodolfo, D. and Jaime, C. (2006), A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operational Research*, **176**(3), 1478-1507.
- Rosenkrantz, D. J., Stearns, R. E., and Lewis, P. M. (1974), Approximate algorithms for the traveling salesperson problem. *Proceedings of the 15th Annual Symposium on Foundations of Computer Science(FOCS 1974)*, 33-42.
- Salhi, S. and Rand, G. K. (1993), Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research*, **66**(3), 313-330.
- Tavakkoli-Moghaddam, R., Safaei, N., and Gholipour, Y. (2006a), A hybrid simulated annealing for capacitated vehicle routing problems with the independent route length. *Applied Mathematics and Computation*, **176**(2), 445-454.
- Tavakkoli-Moghaddam, R., Safaei, N., Kah, M. M. O. and Rabbani, M. (2006b), A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute, In Press, Corrected Proof, Available online 28 February 2006*.
- Timon C. D., Eldon, Y. L. and Defrose, C. (2005), Dynamic vehicle routing for online B2C delivery. *Omega*, **33**(1), 33-45.
- Toth, P. and Vigo, D. (2001), The vehicle routing problem. *Monographs on Discrete Mathematics and Applications*, SIAM, Philadelphia.