

# VRPSPD 해결을 위한 위치기반의 실시간 재경로 탐색 휴리스틱

## A Location-based Real-time Re-routing Heuristic to Solve the VRPSPD

차 상 진<sup>\*</sup>      이 기 성<sup>\*</sup>      유 영 훈<sup>\*\*</sup>      조 근 식<sup>\*\*\*</sup>  
Sang Jin Cha      Kee Sung Lee      Young Hoon Yu      Geun Sik Jo

**요약** 일반적인 Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries (VRPSPD)는 배송과 수거가 동시에 발생하는 문제를 고려한 차량경로 문제이며, 차량의 운행 거리등의 비용을 최소화하는 것을 결정하는 문제이다. 그러나 기존의 VRPSPD는 이미 차량이 출발하기 전에 경로가 정해져 있어서 차량 운행 중 발생하는 고객의 수거 요청을 기존의 경로에 효율적으로 추가하여 서비스하기가 어렵다. 따라서 본 논문에서는 이러한 문제를 위치기반의 서비스를 이용하여 이동 중인 차량의 위치정보를 파악하고 이를 바탕으로 실시간 재경로 탐색을 통해 해결하는 휴리스틱을 제안한다. 그리고 실험을 통해 기존의 방식과 비교하여 차량을 운행하는데 소요되는 비용을 줄이는 결과를 보였다.

**키워드** : 위치기반서비스, 배송 및 수거 차량경로문제, 차량경로문제, 삽입 휴리스틱, 타부 탐색

**Abstract** The vehicle routing problem with simultaneous pick-up and delivery (VRPSPD) is a variant of the vehicle routing problem (VRP) that customers require simultaneously a pick-up and delivery service. The main objective of VRPSPD is to minimize a cost of routes satisfying many constraints. Traditional VRPSPD have been dealt with a static environment. The static environment means that a routing data and plan cannot be changed. For example, it is difficult to change a vehicle's routing plan so that a vehicle serves the pick-up demands of new customers during the delivery service. Therefore, traditional approach is not suitable for dynamic environments. To solve this problem, we propose a novel approach for finding efficient routes using a real-time re-routing heuristics based on the Location Based Service (LBS). Our re-routing heuristics can generate a new route for vehicle that satisfies a new customer's demand considering the current geographic location of a vehicle. Experimental results show that our methodology can reduce the traveling cost of vehicles comparing with other previous methods.

**Keywords** : LBS, VRPSPD, VRP, Insertion Heuristic, Tabu Search

### 1. 서론

인터넷 기술과 서비스의 발전으로 전자상거래 분야가 급속하게 발달하여 국내외의 수송 물동량이 급격히 증대되고 있다. 이로 인해 배송과 수거 서비스를 모두 제공하는 물류 사업이 크게 활성화 되고 있다. 따라서 이러한 물류 사업도 고객에 대한 서비스를 만족시키면서 여러 대의 차량에 대한 총 운행 거리 등의 비용을 최소화 하는데 목적이 있으므로 차량경로문제를 반드시 고려해야 한다[2,7,12].

본 논문에서 다루고자 하는 Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries(VRPSPD)는 일반적인 배송 문제만을 다루는 차량경로문제(Vehicle Routing Problem; VRP) [6, 19]의 변형된 문제로 수거와 배송이 모든 고객에서 일어날 수 있으며, 다양한 고객의 요청이 서로 다른 시간에 발생하는 차량경로문제이다[17]. NP-Hard 문제로 잘 알려진 차량경로문제는 모든 가능한 경로를 조사해보는 완전 최적화 기법(Exact Optimization Method)과 빠른 시간에 근사해를 찾는 메타

<sup>\*</sup>인하대학교 정보공학과 박사과정 aldehyde7@eslab.inha.ac.kr(교신저자), lks@eslab.inha.ac.kr

<sup>\*\*</sup>인하대학교 컴퓨터정보공학부 전임강사 yhyu@eslab.inha.ac.kr

<sup>\*\*\*</sup>인하대학교 컴퓨터정보공학부 교수 gsjo@inha.ac.kr

휴리스틱 기법(Meta Heuristic Method)이 있다[17]. 완전 최적화 기법은 문제에 따라서 최적해를 구하는 시간이 오래 걸리거나, 최적해를 구한다는 보장이 없을 수도 있다. 따라서 실무에서는 최적해는 아니지만 수초 내에 근사해를 찾을 수 있는 메타 휴리스틱 기법을 사용하고 있다. 하지만 이러한 일반적인 차량경로문제에서는 차량이 출발하기 이전에 서비스해야 할 경로가 정해져 있어 새로운 고객 서비스 도중 추가될 경우 재경로 탐색이 어렵다. 여기서 재경로 탐색은 기존의 정해진 차량의 경로를 동적으로 변경하는 것이므로 이미 정해진 경로에 대한 서비스 및 다양한 제약조건을 모두 만족해야만 새로운 고객의 요청을 삽입할 수 있다. 또한 실무에서도 고객들이 요청하는 서비스 시간을 준수하고, 운용비용을 최소화해야 한다. 하지만 운행 중인 차량의 위치를 파악과 다양한 제약조건을 만족하는 경로를 빠른 시간에 찾기가 어려워 차량의 위치 정보를 활용한 재경로 탐색에 대한 휴리스틱이 필요한 실정이다.

따라서 본 논문에서는 이러한 문제를 위치기반 서비스(Location Based Service)를 이용한 실시간 재경로 탐색 휴리스틱으로 해결하고자 한다. 위치기반 서비스는 위치정보를 활용한 서비스로 여러 다른 분야와 접목시켜 부가가치를 높이는 메시업(Meshup) 응용 프로그램으로 많이 연구되고 있다[10,16].

이러한 위치기반 서비스를 이용하여 이동 중인 차량과 실시간으로 추가된 고객의 위치 정보를 파악하고, 삽입 휴리스틱(Insertion Heuristic), 타부 탐색(Tabu Search)을 사용하여 추가된 고객의 서비스 요청을 여러 가지 제약조건을 만족시키며, 차량의 운행시간 비용을 최소로 하는 최적의 경로를 수초 내에 실시간으로 찾아낼 수 있다.

본 논문의 구성은 2장에서 VRPSPD 및 위치기반 서비스에 대한 배경 지식과 관련 연구를 소개한다. 3장에서는 위치기반 서비스를 사용한 실시간 재경로탐색 휴리스틱에 대해 기술하고, 4장 실험 및 평가에서 본 논문에서 제시한 휴리스틱을 실무 데이터를 바탕으로 실험하고 성능을 평가한다. 그리고 5장 결론 및 향후 연구에서 본 논문의 결론과 향후 연구에 대해 언급한다.

## 2. 관련연구

### 2.1 VRPSPD(Vehicle Routing Problem with Simultaneous Pick-ups and Deliveries)

수거와 배송을 고려하는 차량경로문제는 크게 Vehicle Routing Problem with Backhauls(VRPB), Vehicle Routing Problem with Mixed Pick-ups and Deliveries(VRPMPD), VRPSPD로 모델을 나누어 연구되고 있다. 먼저 VRPB는 선-배송 후-수거의 개념으로 차량의 경로를 설정할 때, 배송을 먼저 고려하여 모든 고객에게 배송 서비스를 끝낸 후, 차량의 적재율이 0인 상태에서 수거를 수행하는 단순한 모델이다. 이 모델은 Anily[1]에 의해 비용의 하한값을 사용한 휴리스틱을 사용하여 해결하였고, Brandao[4]는 타부 탐색을 통해 하한값을 통해 나온 해를 개선시키는 방법을 제안 하였다.

그리고 VRPMPD는 배송과 수거가 차량의 경로 안에 순차적으로 혼합된 형태이다. 하지만 배송과 수거가 순차적으로 진행되기 때문에 중량이 다양한 제품에 대해서는 적재용량에 대한 결과가 비효율적인 단점이 있다. 이 모델은 Casco, Golden and Wasil[5]에 의해 배송을 요구하는 고객들에 대해 먼저 경로를 구성한 뒤, 수거 서비스를 요구하는 고객을 삽입하여 최종 경로를 구성하는 연구가 있다.

본 논문에서 다루고 있는 VRPSPD 모델은 VRPMPD보다 좀 더 복잡한 구조를 지닌다. 먼저 배송과 수거가 동시에 모든 고객에서 일어날 수 있다는 가능성을 두어, 순차적이 아닌 난수적으로 존재하게 된다[11]. VRPSPD의 목적은 최소의 비용을 가지는 경로를 알고자 함인데 최소비용은 각 지점 사이의 비용의 합으로 산정된다. 배송 및 수거 차량 경로문제에 추가 되는 제약사항으로는 각 고객의 요청 시간이 제한(Time Constraint)되어 있으며, 차량의 용량(Capacity Constraint)에 초과되는 화물을 실을 수 없다는 조건과 차량의 운행시간이 한정되어 있다는 조건을 가진다[14]. Zachariadis, Tarrantilis and Kiranoudis[17]는 안내지역탐색(Guided local search)과 타부 탐색을 사용한 휴리스틱을 통해 이 모델에 대한 문제를 해결하는 결과를 보였다.

본 논문에서는 다양한 종류의 차량이 존재하며, 수거 서비스를 요청하는 고객이 임의의 시간에 추가되고 이미 서비스를 수행하고 있는 차량의 위치를 기반으로 하여 재경로 탐색을 수행한다. 이러한

문제를 본 논문은 차량과 고객의 위치를 기반으로 하여 삽입 휴리스틱 방법과 타부 탐색을 사용하여 만족 해를 구한다.

## 2.2 위치기반 서비스(Location Based service)

OGC(Open Geospatial Consortium)<sup>1)</sup>에서는 위치 기반 서비스를 이동 통신 기지국과 위성을 이용한 위치 확인 시스템(GPS: Global Positioning System) 등을 통해 파악한 이동 중인 사용자의 위치정보에 의해 작동하는 모든 응용 소프트웨어 서비스라고 정의하고 있다[18, 20].

위성을 이용한 위치 추적 시스템은 이미 교통이나 물류 분야 등에서 많이 사용하면서 그 효용성을 입증한 상태이다. 그러므로 위치기반 서비스가 이러한 위치 추적 시스템과 이동 통신망과 연결되면서 대중적인 서비스로 거듭날 수 있기에 크게 관심을 받고 있다.

또한 여러 인터넷 서비스 제공자들은 위치정보에 대한 OpenAPI를 내놓았으며, 사용자들은 메시업(Mesh up)을 통해 손쉽게 다양한 솔루션들을 제작할 수 있게 되었다[10]. 따라서 기존의 위치기반 시스템에서 플랫폼을 구축하고 실험하는데 있어 지리 정보 시스템(Geographic Information System) 정보 등을 얻기 힘들었으나, 이러한 정보들이 OpenAPI로 제공되어 어려움을 해결할 수 있다. 그래서 위치기반 서비스는 위치 추적뿐 아니라 주변 시설의 정보, 교통, 항법, 전자상거래 등 다양한 형태의 서비스로 활용되고 있다[21].

## 3. 위치기반의 실시간 재경로 탐색 휴리스틱

본 장에서는 1장과 2장에서 설명한 VRPSPD 모델에서의 실시간으로 추가되는 고객의 수거 서비스 요청을 해결하기 위하여 위치기반의 실시간 재경로 탐색 휴리스틱을 제안한다.

먼저 본 논문에서 사용하는 차량 경로 모형에 대한 제약 조건은 다음과 같다.

- 1) 운행되는 차량의 비용은 총 운행 시간에 비례한다.
- 2) 각 차량은 허용된 최대 적재율까지 적재가 가

능하며 모든 차량의 적재율은 동일하다.

- 3) 고객 간의 차량 운행속도는 동일하다.
- 4) 화물의 적재 및 하역에 소요되는 시간은 제품의 무게/부피에 비례하며 운행시간에 포함된다.
- 5) 차량의 출발 및 종착은 거점에서만 이루어지며, 단일 거점만 존재한다.
- 6) 모든 고객 노드들은 배송이나 수거 서비스를 받아야 한다.
- 7) 한 차량의 총 이동거리에 대한 제약은 없다.
- 8) 모든 고객은 서비스가 가능한 시간대를 가지며, 이를 만족해야 한다.
- 9) 차량은 거점에서 동시에 출발하며 거점의 서비스 종료 시간 내에 복귀해야 한다.

따라서 목적함수에 해당하는 제약조건 1)은 전체 차량 운행 시간의 최소화를 의미한다. 목적함수는 다른 제약조건을 모두 만족하며, 결과를 도출해야 한다.

제안하는 휴리스틱을 적용한 시스템의 구조는 그림 1과 같이 크게 위치기반 서비스를 이용하여 차량과 고객의 위치를 알아내는 모듈과 이를 바탕으로 재경로를 탐색하는 모듈로 구성되어 있다.

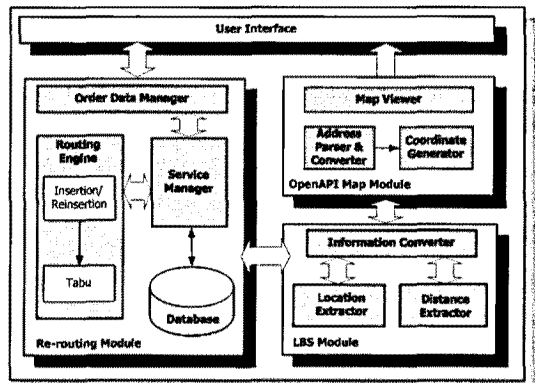


그림 1. 시스템 구조

### 3.1 위치기반 서비스 모듈

위치기반 서비스 모듈의 위치 추출기(Location Extractor)에서 이동 중인 차량의 위치를 운전자의 휴대 단말기를 이용한 위치 탐색 방법을 사용하여 알아낸다.

이동 중인 차량의 위치는 무선망과 사용자의 휴대 단말기 등의 위치 측위 장비 및 소프트웨어를 이용하여 측정하고 정보 변환기(Information Converter)

1) Open Geospatial Consortium;  
<http://www.opengeospatial.org>

에서 위치를 표시하는 지도에서 사용하는 위경도로 변환하게 된다.

수거 서비스를 요청하는 고객의 위치는 OpenAPI 지도 모듈의 주소 변환기(Address Parser & Converter)를 사용하여 고객의 주소 정보를 바탕으로 위경도 생성기(Coordinate Generator)를 사용하여 알아낸다. 이렇게 알아낸 차량과 고객의 위치 정보를 바탕으로 거리 추출기(Distance Extractor)를 사용하여 차량경로를 수립하기 전에 서비스를 해야 하는 모든 고객 및 거점(Depot)에 대한 최단 경로인 거리정보를 먼저 구하게 되고 이를 바탕으로 차량경로를 수립하게 된다. 이 때 거리정보는 계속 시스템에서 유지하고 있어야하며, 서비스를 시작한 이후 새로 추가되는 고객의 위치 및 차량의 위치가 추가되면 이에 대한 새로운 거리정보도 추가하여 재경로를 탐색할 때 이용하게 된다.

### 3.2 재경로 탐색 모듈

앞 절에서 소개한 위치기반 서비스 모듈에서 측정 한 차량과 고객의 위치 정보 및 거리정보를 기반으로 본 절에서는 추가되는 고객의 서비스 요청을 처리하는 재경로 탐색 모듈을 소개한다.

본 논문에서 제안하는 재경로 탐색 휴리스틱은 거점에서부터 출발한 차량들이 각 군집에서 서비스를 수행하고 있을 때, 새로운 고객이 수거 서비스를 요청하게 되면 위치기반 서비스 모듈로부터 차량 및 고객의 위치 및 거리를 파악하게 되고 이를 바탕으로 여러 군집들의 앞으로 서비스해야 할 경로들 중에서 가장 비용이 적게 드는 경로에 고객을 삽입하는 기법이다.

#### 3.2.1 재경로 탐색 절차

재경로 탐색은 삽입 휴리스틱(Insertion Heuristic) [13]과 타부 탐색(Tabu Search)[8,9,15]을 사용하여 구해진 초기 경로를 가지고 각 차량이 서비스를 이미 시작된 상태라고 가정을 한다. 그리고 이미 차량이 서비스를 시작한 상태에서 새로운 고객의 수거 요청이 들어왔을 경우 차량의 위치와 서비스 제공 받지 못한 고객과 새로 추가된 고객을 가지고 재경로 탐색을 수행한다. 재경로 탐색의 절차는 그림 2와 같다.

[단계]

단계1. 초기해를 생성한다. 초기해는 삽입 휴리스

틱을 사용하여 구한다.

단계2. 초기해를 바탕으로 타부 탐색을 사용하여 단계1에서 나온 해를 다시 개선하여 서비스 할 초기해를 갱신한다. 이 결과를 가지고 최초의 차량의 운행이 시작된다.

단계3. 새로운 고객의 요청이 들어온 시점의 각 차량( $v$ )의 위치정보와 현재까지 서비스 하지 않은 모든 고객( $r_{ij}$ )의 정보를 위치기반서비스 모듈(LBS Module)을 통해 파악하고 관련된 거리정보를 추출한다.

단계4. 아직 서비스 하지 않은 모든 고객( $r_{ij}$ )에 추가 된 고객( $k$ )을 모두 삽입하여 보고 시간제약조건과 차량의 적재조건에 위배되지 않으며 가장 이동시간 비용( $c$ )이 적게 증가 되는 경로를 찾는다.

단계5. 선택된  $r_{ij}$ 를 삭제하고 그 위치에 새로운 경로인  $r_{ik}, r_{kj}$ 를 추가한다.

단계6. 추가 되는 고객이 발생 시 단계3으로 이동한다.

Procedure LBS\_Insertion(Customer k, ArrayRoute R)

```

R=Insertion_Algorithm(k, R)
for each exist route r in R
    r=Tabu_Search(r)
Start routing plan based R
While exist the customer which do not served
    if exist new customer k then
        Verify location of each truck and served customers
        by LBS Module
        Create a array with row i correlative customer i,
        and column j correlative customer j
        for each exist  $r_{ij}$ 
            if route which inserted k to  $r_{ij}$  satisfy time and
            capacity constraints then
                 $c = t_{ik} + t_{kj} - t_{ij}$ 
                input c to array[i][j]
            else
                input infinity to array[i][j]
        //end of for each
        find index i and j that have minimum value in array
        delete existing edge  $r_{ij}$ 
        add new edge  $r_{ik}$  and  $r_{kj}$ 
    else
        continue;
//end of While
End of Procedure
    
```

그림 2. 재경로 탐색 절차

### 3.2.2 삽입 휴리스틱(Insertion Heuristic)

시간제약이 있는 차량경로문제의 해법으로 잘 알려진 삽입 휴리스틱은 차량의 초기 경로를 생성하는 방법으로 Solomon[13]에 의해 처음으로 소개되었다. 삽입 휴리스틱은 새로운 경로에 빠른 시간에 고객을 삽입하는데 효과적인 방법으로 알려져 있다 [3]. 그림 3은 삽입 휴리스틱의 절차를 보여준다. 그림 3과 같이 할당되지 않은 고객을 기존에 존재하는 경로에 삽입해 보면서 제약조건들을 만족하는지 확인하고, 만약 제약조건을 만족하지 않으면 새로운 경로를 생성하면서 서비스할 고객을 모두 삽입하여 결과를 보여준다.

```

Procedure Insertion_Algorithm(Customer k,
ArrayRoute R)

While exist unassigned customer do
//Have all the customers been served?
  Create a array with row r correlative route, and
  column k correlative customer
  For each unassigned customer k
    For each exist route r in R
      input best inserted cost of customer k
      to array[r][k]
      remember the position inserted
    If exist a customer k can not be assigned any
    route
      then
        R Select an unused vehicle and create a new route
    Else
      Try to find the a customer and route pair that
      has the minimal C cost
      Insert the selected customer into the route
//end of While
Return R
End of Procedure

```

그림 3. 삽입 휴리스틱 절차

본 논문에서는 초기 경로 생성에서는 일반적인 삽입 휴리스틱을 사용하지만, 차량이 서비스를 시작한 상황에서는 차량별로 이미 서비스를 완료한 고객을 제외한 남은 고객에 새로 수거 서비스를 요청하는 고객을 삽입하여 경로를 생성하는 변형된 방법을 사용한다.

### 3.2.3 타부 탐색(Tabu Search)

메타 휴리스틱의 하나인 타부 탐색 기법은 메모리 기반 탐색 전략으로서(Memory Search Based Search Strategy) 국지 최적 해에서 머무르지 않고

바람직한 해를 구하도록 제안된 방법이다[8,9,15]. 타부 탐색은 Glover[8,9]에 의해 정립되었고 짧은 시간 안에 해를 구하기 힘든 많은 문제에 응용되고 있다.

그림 4는 타부 탐색의 절차를 보여준다. 그림 4와 같이 초기해( $i$ )로부터 탐색을 시작하여 이를 현재해로 설정을 한다. 그리고 이웃 해  $N(i)$ 에 대하여 탐색을 하고 그 중에 목적함수( $f$ )가 가장 많이 개선되거나, 개선이 되는 경우가 없을 시에는 가장 목적함수의 증가가 적은 값을 갖는 것이 새로운 현재해가 된다. 현재해를 기준으로 계속적으로 탐색을 하여 미리 정해 놓은 반복 횟수에 도달하거나 해의 개선 정도가 일정수준 이하로 떨어질 경우 탐색을 종료하게 된다.

```

Procedure Tabu_Search(Initial_Solution i)

i* = i
k = 0
While stopping condition is false do
  k = k + 1
  Generate a subset V* of solution in N(i, k) such that
  either one of the Tabu conditions is violated or at least
  one of the aspiration conditions holds.
  i = best solution in V*
  If f(i) < f(i*) then
    i* = i
  Update Tabu and aspiration conditions
//end of While
Return I*
End of Procedure

i: solution indexes
k: iteration index
V*: subset of solution
N(i, k): neighborhood of solution i at iteration k
f(i): objective function value for solution i

```

그림 4. 타부 탐색 절차

타부탐색은 지역 최적해를 알아낼 수는 있으나 전체 최적해를 보장할 수 없다는 한계점을 지니기 때문에 메모리 전략을 이용한 타부목록과 열망수준(Aspiration Level)을 사용하여 전체 최적해를 보장할 수 없는 부분을 보완하고 있다. 타부목록의 크기는 타부목록에 저장되는 타부속성의 개수를 나타내는 것으로 타부목록의 크기만큼 해의 순환을 제약하게 된다. 이 목록의 크기가 너무 작으면 순환(Cycle)에 빠지기 쉽고 반대로 너무 크면 순환은 방지할 수 있으나 지역 최적해에 빠지기 쉬워진다.

따라서 적당한 크기의 목록을 실험을 통해 정하게 된다. 순환을 방지하기 위한 타부 제약은 오히려 좋은 해가 있을 수 있는 곳으로의 이동을 방해하기도 한다. 이러한 한계점을 보완하기 위해 새로운 조건으로 열망수준을 정하게 된다. 열망수준은 새로운 가능성이 있는 해가 타부라 하더라도 지금까지 발견된 좋은 해보다 더 좋은 경우에 타부를 무시하고 새로운 해로 받아들이는 경우로 설정하게 된다.

4. 실험 및 평가

본 논문에서 제안한 위치 기반의 실시간 재경로 탐색 휴리스틱을 실험하기 위하여 실제 국내 H사의 2009년 9월 물류 데이터 사용 하였다. 또한 전자지도를 구축하기 위해 메시업으로 OpenAPI로 제공되는 네이버 지도<sup>2)</sup>를 실험에 사용하였다.

먼저 실험에 사용된 거점 및 고객의 위경도 값은 그림 5와 같이 주소 변환기와 위경도 생성기를 통해 구하였다. 그림 5는 표 2의 고객 '100195'의 주소를 위경도로 변환한 것이다. 따라서 차량의 위치 정보는 GPS를 통해 얻을 수 있으며, 고객의 위치 정보는 주소를 통해 얻을 수 있다.

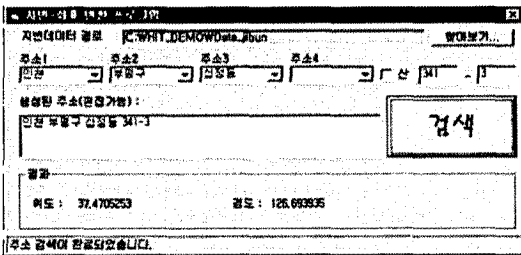


그림 5. 주소를 통한 위경도 변환

표 1. 거점 정보

ID	Latitude	Longitude	Ready Time	Due Time	Demand
RD22	37.452826	126.620069	0830	1900	0

표 1은 거점에 대한 위치 및 운영 시간을 나타낸 것이다. 표 1의 거점의 운영 시간을 바탕으로 차량은 이 시간 이내에 모든 서비스를 완료하고 거점으로 복귀 하여야 한다.

표 2. 고객 주문 정보

ID	Latitude	Longitude	Ready Time	Due Time	Demand
100195	37.470525	126.693935	0930	1030	-90
100272	37.571017	126.915544	1200	1500	-22
100393	37.446409	127.001473	1200	1400	+17
100456	37.381608	126.761025	1000	1100	-81
100460	37.447705	126.794543	1130	1200	+11
100462	37.466026	126.667072	1200	1300	-61
100474	37.407233	126.728714	0930	1030	+24
100478	37.542279	126.671804	1130	1200	+21
100484	37.457132	126.702691	0900	1000	-26
100506	37.679359	126.780498	1000	1100	-22
100513	37.691442	126.746803	1030	1200	+76
100514	37.654475	126.627586	1300	1500	-2
100521	37.615348	126.726387	1300	1600	+15
100531	37.637019	126.715953	0900	1100	-10
101754	37.327631	126.78536	1030	1200	-145
102000	37.6567	126.831736	1100	1400	-1
102027	37.4669	126.692845	0900	1000	+11
200015	37.494883	126.894354	1030	1130	+1
200016	37.640917	126.832076	1100	1300	-18
200017	37.666442	126.800722	1100	1200	+20
200049	37.506398	126.713513	1100	1200	-120

표 2는 실험 데이터의 특징일에 대한 초기 차량 경로를 구성할 때 필요한 고객의 주문 정보이다. 거점 정보와 마찬가지로 위치 및 서비스를 받을 시간이 설정되어 있으며, 요구되는 수량은 양의 값은 수거를 의미하며, 음의 값은 배송을 나타낸다. 여기서 서비스할 제품은 평균값으로 중량은 13kg이며, 용량은 0.065CBM으로 계산 하였다. 실험에 사용한 차량은 5t 2 대(8064, 8041), 2.5t 3대(8059, 8035, 8040)로 구성되어 있으며, 주문량에 따라 주문이 적은 경우 주어진 차량수 이내에서 서비스가 이루어지고, 주문이 많은 경우 주어진 차량 외에 용차(5t)를 사용할 수도 있다. 차량의 평균 이동 속도는 50km/h로 같으며, 차량의 용량은 각각 26CBM(5t), 13CBM(2.5t)이고 차량의 중량이나 용량을 초과하여 제품을 적재할 수 없다.

표 2의 고객 주문 정보와 이를 바탕으로 거리 정보를 구한 다음 차량이 서비스를 수행해야 할 초기 경로를 구성하게 된다. 그림 6과 표 3은 표 2의 고객 주문 정보를 바탕으로 얻어진 결과를 나타낸 것

2) NaverOpenAPI, NHN Corp.  
http://openapi.naver.com

이다. 그림 6에서 차량의 이동 경로는 실제 도로를 통해 이동하지만, 개념적인 이동 순서를 파악하기 위해 직선으로 표기하였다. 그림 6과 표 3의 초기 경로를 바탕으로 차량들은 서비스를 수행하게 된다.

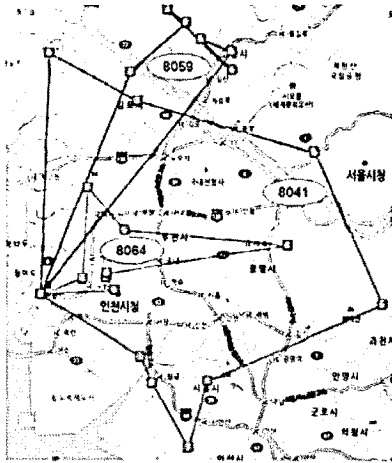


그림 6. 초기 경로 구성결과

표 3. 초기 경로 결과

ID	Vehicle	Order	Arrival	Departure
RD22	8041	0	-	0930
100474	8041	1	0950	0957
100456	8041	2	1011	1032
101754	8041	3	1045	1122
100460	8041	4	1131	1134
100393	8041	5	1203	1208
100272	8041	6	1247	1253
100521	8041	7	1321	1325
100514	8041	8	1340	1341
RD22	8041	-	1420	-
RD22	8059	0	-	0930
100531	8059	1	1015	1021
100506	8059	2	1032	1038
100513	8059	3	1044	1104
200016	8059	4	1118	1123
200017	8059	5	1124	1130
102000	8059	6	1134	1135
RD22	8059	-	1236	-
RD22	8064	0	-	0930
100484	8064	1	0944	0951
102027	8064	2	0954	0957
100195	8064	3	1001	1024
200015	8064	4	1046	1047
200049	8064	5	1111	1142
100478	8064	6	1151	1157
100462	8064	7	1210	1226
RD22	8064	-	1237	-

차량이 서비스를 수행하는 동안 표 4의 '102029' 고객이 그림 7과 같이 삽입되게 된다. 삽입되는 시간은 해당 고객의 서비스 요청 시간에서 시작 시간 (Ready Time)으로 설정하고, 그림 7과 같이 이때의 차량의 위치를 파악하여 차량이 앞으로 서비스해야 할 고객과 새로 추가된 고객을 바탕으로 제약조건을 만족하는 비용이 최소가 되는 경로에 고객을 삽입하게 된다.

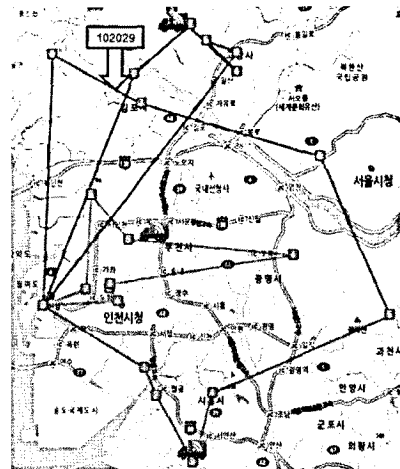


그림 7. 첫 번째 고객 삽입

표 4. 추가되는 고객 주문 정보

ID	Latitude	Longitude	Ready Time	Due Time	Demand
102029	37.617753	126.696703	1100	1300	+16
200021	37.510667	126.773536	1300	1500	+17

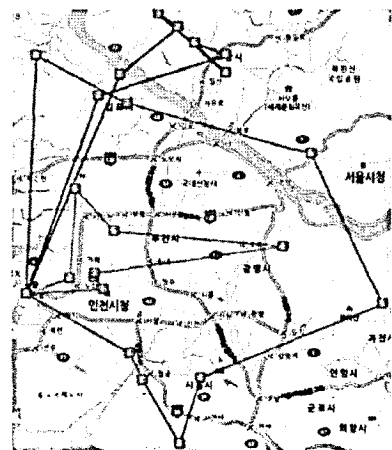


그림 8. 첫 번째 고객 삽입 결과

그림 8에서 보이는 결과처럼 '102029' 고객은 8059 차량에 삽입되었다. 실제 업무에서는 차량이 거점으로 돌아온 다음에 삽입된 고객의 서비스를 수행하기 위해 다시 차량이 출발하거나, 근처에서 업무를 마치고 복귀하는 차량이 서비스를 수행하게 되는데, 여기에서는 거점과 '102029' 고객 사이의 최단 거리가 29.923km로 차량의 평균 속도로 계산하면 거점 복귀 후 고객을 방문하기에는 서비스 요청 시간 제약 조건을 지킬 수 없다. 따라서 운행 중인 차량의 남은 고객과 추가된 고객을 바탕으로 재경로를 탐색한 결과 8059 차량에 고객이 삽입됨을 볼 수 있다.

그리고 표 4의 두 번째 고객 '200021'이 삽입되었을 때의 고객 위치 및 차량의 위치는 그림 9와 같다. 첫 번째 고객과 마찬가지로 고객의 위치 및 차량의 위치를 기반으로 제약조건을 만족하는 경로 중 비용이 가장 적게 소요되는 경로를 선택하게 된다. 이 경우에는 두 대의 차량이 이미 서비스를 마치고 거점으로 복귀된 상태이고 나머지 한 대의 차량은 서비스를 수행 중인 상태이다. 따라서 고객 '200021'을 모든 차량이 서비스를 수행 할 수 있으나, 이동 거리에서 8041 차량이 서비스를 수행할 경우 28.238km의 거리가 절약되어 8041 차량의 경로에 그림 10처럼 삽입이 됨을 확인할 수 있다.

또한 얻어진 결과를 바탕으로 운행시간을 비교해 보았다. 비교 대상은 먼저 추가되는 고객을 초기에 주문 정보에 모두 포함되었을 경우를 가정하여 산출된 결과와 실무와 같이 모든 차량이 초기에 정해

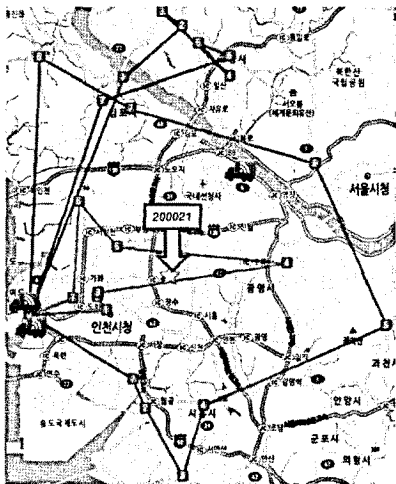


그림 9. 두 번째 고객 삽입

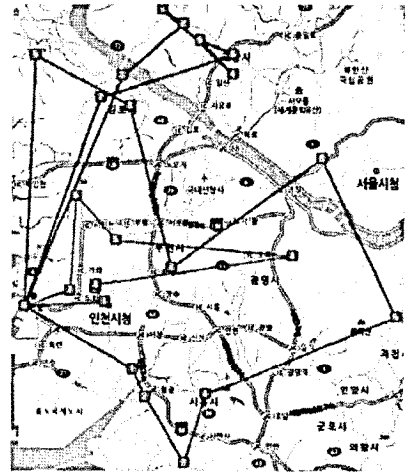


그림 10. 두 번째 고객 삽입 결과

진 경로를 운행하고 거점 복귀 후 추가된 고객의 수거 서비스를 수행하는 방법과 본 논문에서 제안한 방법의 운행시간이다. 운행 시간에 대한 비교 결과는 표 5와 같다. 표 5와 같이 모든 고객에 대한 주문을 포함하여 경로를 구성한 결과보다는 실무 방법과 제안한 방법 모두 운행시간이 증가함을 알 수 있다. 이는 경로 구성에 있어 차량의 이동성을 고려하지 않은 최적 근사해로 비교의 기준이 된다. 제안한 방법은 기존의 실무 방법에 비해 총 19분의 운행시간 단축을 보였다.

표 5. 기존방법들과의 운행시간 비교

Vehicle	Include All Orders (min)	Return to Depot (min)	Proposed (min)
8041	339	355	403
8059	213	261	252
8064	301	309	251
Total	853	925	906

표 6은 앞에서 보인 실험 결과를 포함한 2009년 9월의 전체 물류 데이터에 대한 동일한 실험에 따른 비교를 보여준다. 제안된 방법은 실무에서 사용하는 방법에 비해 평균적으로 약 2.16%의 운행시간 절감 효과를 보였다.



표 6. 전체 물류 데이터의 운행시간 비교

Date	Include All Orders (min)	Return to Depot (min)	Proposed (min)
0903	853	925	906
0904	1,706	1,850	1,815
0905	1,436	1,557	1,525
0906	1,665	1,805	1,765
0908	809	877	860
0909	2,677	2,900	2,840
0910	2,400	2,602	2,541
0911	1,769	1,918	1,870
0912	2,633	2,855	2,796
0916	1,048	1,140	1,111
0917	2,627	2,848	2,790
0918	2,220	2,410	2,358
0919	2,015	2,185	2,136
0920	1,907	2,068	2,025
0922	935	1,079	1,056
0923	1,968	2,135	2,091
0924	1,687	1,829	1,787
0925	1,651	1,800	1,752
0926	2,018	2,159	2,143
0927	2,020	2,200	2,145
0929	1,896	2,055	2,013
Total	38,000	41,197	40,325

## 5. 결론 및 향후 연구

수거와 배송이 동시에 일어나는 VRPSPD 모델에서 이미 차량이 서비스를 시작한 후, 새로운 고객의 수거 서비스 요청이 발생하였을 때, 기존의 모델은 중간에 경로를 변경하기가 어려워 근처에서 서비스를 수행하는 차량이 수거하거나 거점으로 서비스를 마치고 돌아온 차량이 다시 출발하여 수거를 담당했으나 본 논문에서는 이동 중인 차량 및 고객의 위치를 기반으로 실시간으로 재경로 탐색을 수행함으로써 다양한 제약조건을 만족시키며 서비스를 수행할 수 있으며 기존 모델과 비교를 통해 차량의 운행시간을 단축하는 결과를 보였다.

여기서 차량의 위치는 GPS를 통해 얻을 수 있었으며, 고객의 위치 정보는 주소를 변환하여 얻을 수 있었다. 이러한 위치기반의 정보를 바탕으로 Open-API로 제공하는 지도 서비스를 사용하여 본 논문에서와 같이 차량 경로문제로 매시업하여 새로운 응용 어플리케이션을 생산하는 효과를 가져왔다.

그러나 위치기반 서비스에서 단지 위치와 관련된 정보만을 사용하므로 교통 정보와 같은 추가적인 정보를 사용하지 못하는 한계점이 있었다. 이러한 문제는 교통 상황에 따라서 서비스 시간이 지연되거나 단축되는 경우 새로운 고객의 삽입과 더불어 남은 경로에 대해 운행 비용을 절약할 수 있는 여지가 있다. 따라서 고객을 추가적으로 삽입할 때 실제 도로 간의 교통 상황까지 고려한 향상된 휴리스틱에 대한 연구가 필요하다.

## 참고 문헌

- [1] Anily, S., 1996, "The vehicle routing problem with delivery and back-haul options," *Naval Research Logistics*, vol. 43, pp.415-434.
- [2] Antoine L., Y. Mati, Z. Binder, 2001, "A tabu search heuristic for the single vehicle pickup and delivery problem with time windows," *Journal of Intelligent Manufacturing*, vol. 12, pp.497-508.
- [3] Baker B. D., V. Furnon, P. Shaw, 2000, "Solving vehicle routing problems using constraint programming and meta heuristics," *Journal of Heuristics*, vol. 6, pp.501-523.
- [4] Brandao, J., 2006, "A new tabu search for the vehicle routing problem with back-hauls," *European Journal of Operational Research*, vol. 173, no. 2, pp.540-555.
- [5] Casco, D. O., B. L. Golden and E. A. Golden, 1988, "Vehicle Routing with Backhauls: Models Algorithms, and Case Studies," *Vehicle Routing: Methods and Studies*, pp.127-147.
- [6] Clark, G. and J. K Wright, 1964, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, vol. 2, no. 4, pp.568-581.
- [7] Gendreau M., G. Laporte, D. Vigo, 1999, "Heuristics for the traveling salesman problem with pickup and delivery," *Computers & Operations Research* vol. 26, pp.699-714.
- [8] Glover, F., 1989, "Tabu Search Part I", *ORSA Journal on Computing*, vol. 1, no. 3, pp.190-206.
- [9] Glover, F., 1990, "Tabu Search Part II", *ORSA*

- Journal on Computing, vol. 2, no. 1, pp.4-32.
- [10] Meshup: [http://en.wikipedia.org/wiki/Meshup\\_\(web\\_application\\_hybrid\)](http://en.wikipedia.org/wiki/Meshup_(web_application_hybrid))
- [11] Min, H., 1989, "The multiple vehicle routing problem with simultaneous delivery and pick-up," *Transportation Research Part A*, vol. 23A, no. 5, pp.377-386.
- [12] Quan L., M. M. Dessousky, 2006, "A new insertion-based construction heuristic for solving the pickup and delivery problem with time windows," *European Journal of Operational Research* vol. 175, pp.672-687.
- [13] Solomon M.M., 1987, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, pp.254-265.
- [14] Tan K.C., L.H. Lee, Q.L. Zhu, K. Ou, 2001, "Heuristic methods for vehicle routing problem with time windows," *Artificial Intelligence in Engineering*, vol. 15, pp.281-295.
- [15] William P. Nanry, J., W. Barnes, 2000, "Solving the pickup and delivery problem with time windows using reactive tabu search," *Transportation Research Part B*, vol. 34, pp.107-121.
- [16] Yu Y.H., J.H. Kim, K.C. Shin, G.S. Jo, 2009, "Recommendation system using location-based ontology on wireless internet: An example of collective intelligence by using 'mashup' applications," *Expert Systems with Applications*, Vol 36, Issue 9, pp.11675-11681.
- [17] Zachariadis E.E., C.D. Tarantilis, C.T. Kiranoudis, 2009, "A hybrid metaheuristic algorithm for the vehicle routing problem with simultaneous delivery and pick-up service," *Expert Systems with Applications*, vol 36, issue. 2, pp.1070-1081.
- [18] 문형돈, 이성휘, 이재환, 2004, "국내 LBS 동향 및 활성화 방안 분석," *정보통신연구진흥원 학술정보 주간기술동향* 1133호.
- [19] 유영훈, 차상진, 조근식, 2009, "시간 제약을 가지는 차량 경로 스케줄링 문제 해결을 위한 기회시간 반영 하이브리드 휴리스틱," *지능정보연구*, 제 15권, 제3호, pp.129-142.
- [20] 조대수 2007, "특허 분석을 통한 LBS 기술현황에 관한 연구," *한국공간정보시스템학회논문지*, 제9

권, 제3호, pp.65-75.

- [21] 진희채, 홍길표, 2003, "위치기반 서비스를 위한 모델 분석 및 기술 연구," *한국GIS학회 춘계학술대회*, pp.438-450.

논문접수 : 2010.05.25

수정일 : 1차 2010.08.04 / 2차 2010.08.09

심사완료 : 2010.08.10



#### 차 상 진

2006년 인하대학교 컴퓨터공학 학사  
2009년 인하대학교 대학원 정보공학과 석사  
2009년~현재 인하대학교 대학원 정보공학과 박사과정

관심분야 : Semantic Web, Vehicle Routing Problem, Knowledge Management, Collective Intelligence, Web Service



#### 이 기 성

2003년 백석대학교 컴퓨터공학 학사  
2005년 인하대학교 대학원 컴퓨터정보공학과 석사  
2005년~2008년 삼우통신(주) 연구원  
2005년~현재 인하대학교 대학원 정보공학과 박사과정

관심분야 : Semantic Web, Semantic Image retrieval, Augmented Reality



#### 유 영 훈

2000년 인하대학교 대학원 전자계산공학 석사  
2001년 (주)소프트웨어벨리 대표  
2010년 인하대학교 대학원 정보공학과 박사

2008년~현재 인하대학교 컴퓨터정보공학과 전임강사

관심분야 : Constraint Satisfaction Problems, Resource Optimization, Meta-Heuristic, AI



#### 조 근 식

1982년 인하대학교 전자계산학 학사  
1985년 Queens College, CUNY M.A.  
1991년 City University of New York Ph.D.  
1991년~현재 인하대학교 컴퓨터정보공학과 교수

2006년~현재 BK21 지능형 유비쿼터스 물류 기술 연구사업단장

관심분야 : AI, Semantic Web, Intelligent Agent System