

# Pickup Point 최적입지선정을 위한 Greedy Heuristic Algorithm 개발 및 적용: 서울 대도시권 지하철 시스템을 대상으로

박종수\* · 이금숙\*\*

**요약:** 서울 수도권 지하철 승객이 인터넷으로 구매한 채소 등의 신선 식품을 퇴근하는 도중에 이동 경로상의 지하철역 구내의 서비스 시설물에서 배달받는다 가정하면, 제기되는 문제는 어떤 지하철역에 서비스 시설물을 설치하면 얼마나 많은 승객들이 이용하는 가를 파악하는 것이다. 이러한 문제는 픽업 문제로 알려져 있고, 픽업 문제를 풀려면 교통 흐름을 설명하는 교통 네트워크가 있어야 되고 각 교통 흐름의 이동 경로도 알 수 있어야 된다. 서울 수도권 교통카드 시스템의 교통카드 트랜잭션 데이터베이스로부터 지하철 승객의 출발역에서 도착역 까지 움직이는 이동 경로인 승객 흐름을 구할 수 있으므로, 서울 수도권 지하철 교통 시스템에서 픽업 문제는 승객 흐름을 최대한 많이 포착하는 서비스 시설물들을 설치할 지하철역들을 선정하는 것이다.

본 논문에서 서울 수도권 지하철 시스템과 지하철 승객들의 흐름을 픽업 문제의 모델에 맞추어 수식화하였고, 이를 기반으로 하여 지하철 승객들의 흐름을 기종점 통행 행렬을 만들어 단계별로 가장 많은 통과 승객수를 갖는 픽업 역을 계산해 내는 빠른 휴리스틱 알고리즘을 제안하였다. 대용량 교통 네트워크와 교통 흐름에 해당되는 약 400여 개의 지하철역을 가진 서울 수도권 지하철 네트워크와 약 500만 명의 지하철 교통카드 트랜잭션들을 입력으로 하여 제안된 알고리즘으로 선정된 픽업 역들을 빠른 시간 내에 얻었고, 그 중에서 상위 10개 픽업 역들을 지하철 안내 지도에 표시하여 설명하였다. 그리고, 그 결과가 거의 최적해와 같음을 추가 실험을 통하여 검증하였다.

**주요어:** 승객 흐름, 교통 네트워크, 픽업 문제, 흐름 포착 위치 할당 모델(FCLM), 휴리스틱 알고리즘, 최적화 문제, 서울 수도권 지하철 시스템

## 1. 서론

인터넷과 전자 상거래의 발전으로 고객이 원하는 물품을 인터넷으로 주문하여 배달받는 시스템이 활성화되어 있다. 여러 물품 중에서도 고객이 직접 지정하는 판매점이나 서비스 장소에서 물품을 구매하

거나 전달받기를 원하는 경우도 많다. 예를 들면, 채소나 생선 등의 신선 식품을 구매하는 고객은 냉장 보관되어 있는 서비스 장소에서 직접 전달받아 가는 것이 좋은 경우이고, 전기 자동차의 급속 충전 시스템을 이용하는 고객은 서비스 위치로 가서 그 시설을 사용하는 경우이다. 이와 같은 서비스 시설물은 가능

이 논문은 2010년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.

\* 성신여자대학교 IT학부 교수

\*\* 성신여자대학교 지리학과 교수

한 많은 고객들이 편리하게 이용할 수 있는 위치에 설치되어 있어야 고객의 편의성과 경제의 활성화에 큰 기여를 할 수 있다. 고객들의 이동 경로에 따라 영향을 받는 시설물들의 종류로는 편의점, 자동차 검사소, 광고 전광판, 주유소 등이 추가 된다(Hodgson, 1990; Zeng, *et al*, 2009; Linesa, *et al*, 2008; Hornera & Groves, 2007; Kuby and Li, 2005; ReVelle and Eiselt, 2005; Hodgson, *et al*, 1996).

경로와 위치 개념이 결합된 추상 모델은 네트워크로 표현할 수 있고, 이 네트워크에서 고객이나 자동차의 흐름이 있으면 특정 위치의 시설물을 통과하는 흐름의 양을 측정할 수 있다. 네트워크에서 흐름을 포착해내는 모델로는 p-median model, maximal covering location model, 흐름 포착 위치 할당 모델(flow capturing location allocation model, FCLM)이 알려져 있고, 최근에는 고객의 위치 선호도를 수치화한 모델로 benefit-maximizing pickup location-allocation model이 있다(Hodgson, 1990; Zeng, *et al*, 2009). FCLM 모델에 적용하는 자동차나 승객들의 흐름에 대한 실제 세계에서 OD(Origin - Destination) 흐름과 그 경로들을 구하기는 무척 어렵다(Hodgson, 1990; Zeng, *et al*, 2009). 그러나 서울 수도권 교통카드 시스템에서는 지하철 승객들의 OD 흐름과 그 이동 경로를 구할 수 있어서(박중수 · 이금숙, 2010; Lee, *et al*, 2008, 2011) 대규모 네트워크와 대용량의 승객 흐름을 입력으로 하는 실제 문제에 적용할 수 있다. 수도권 지하철을 사용하는 승객들 중에서 인터넷으로 상품을 구매하여 집으로 퇴근하는 도중에 그 상품을 직접 전달받는 것을 원할 수 있다. 상품 택배를 신청할 때 집에서 그 것을 받을 사람이 없거나 또는 신선한 채소 등을 냉장 보관소에서 직접 전달을 받기를 원하는 경우(pickup)가 많을 것이다. 이런 상황에서 수도권 지하철을 사용하는 승객들을 위한 상품의 전달 장소로 지하철역 내부에 픽업 서비스 시설물을 설치하는 경우라면, 수도권 지하철역들 중에서 어느 역에 서비스 시설물을 설치하느냐가 중요한 문제가 되고 그리고 몇 개를 설치하면 전체 승

객들이 얼마나 많이 이용 가능할 것인가 하는 문제도 중요하게 된다. 본 논문에서는 FCLM에 기반으로 서울 수도권 지하철 시스템의 네트워크에 지하철 승객들의 이동 흐름을 포착해내는 픽업(pickup) 문제를 해결하는 방법론을 서술한다.

픽업 문제는 노드들인  $n$ 개의 지하철역들로 구성된 네트워크에서 지하철 승객들의 승객 흐름을 최대한 많이 포착하게 하는  $p$ 개의 pickup 지하철역을 선정하는 것이다. 이 문제는 NP-hard 문제로 알려져 있어서(Hodgson, 1990), 문제를 해결하기 위해 주로 상업용 최적화 소프트웨어 패키지(예, ILOG CPLEX 12.2)를 사용하여 선형 및 정수 최적화 모델을 이용하여 픽업 문제나 FCLM 문제를 해결한다. 이런 경우에도 대부분의 입력의 개수가 작은 한정된 모델에 해당되는 네트워크에서 교통 흐름에 대한 분석을 수행해 오고 있다. 본 논문에서 고려되고 있는 교통 네트워크인 서울 수도권 지하철 시스템 상의 노드의 수  $n$ 은 약 400개로 아주 큰 네트워크에 해당되고 약 500만 명의 지하철 승객의 승객 흐름을 이러한 상업용 최적화 소프트웨어의 입력으로 사용하기에는 너무 큰 양의 데이터가 된다. 대용량의 입력으로 선형 최적화 문제가 훨씬 더 복잡하고 실행 시간이 무척 오래 걸릴 수 있기 때문에 최적의 해답(optimal solution)을 구하는 방법 대신에 빠른 시간 내에 거의 최적에 가까운 해답(near-optimal solution)을 구할 수 있는 heuristic 알고리즘을 모색하는 것이 본 논문의 연구 초점이다.

본 논문에서 서울 수도권 지하철 시스템과 지하철 승객들의 흐름을 픽업 문제의 모델에 맞추어 수식화하였고, 이를 기반으로 하여 지하철 승객들의 흐름을 기종점 통행 행렬(O-D trip matrix)을 만들어 단계별로 가장 많은 통과 승객수를 갖는 픽업 역을 계산해내는 휴리스틱 알고리즘을 제안하였다. 약 500만 명의 수도권 지하철 승객들의 트랜잭션들을 분석하여 제안된 알고리즘으로 픽업 역을 계산해내는 실험 결과를 보여주고 거의 최적의 결과임을 다른 관점에서도 분석하였다. 서울 수도권 지하철 시스템에 관한

여러 각도에서의 분석(박종수 · 이금숙, 2010; Lee, *et al*, 2008, 2011)은 승객들의 통행 특성을 이해하고, 수도권 지하철 교통망을 보다 효과적으로 이용할 수 있는 방안을 연구할 수 있고, 그리고 수도권 대중교통의 정책 입안에 도움을 줄 수 있다.

본 논문의 구성에 대해 설명하면, 2절에서는 서울 수도권 지하철 시스템에서 승객흐름을 포착하는 픽업 문제에 대한 모델을 설명하고, 3절에서는 모델의 답을 구하기 위한 greedy heuristic 알고리즘에 대한 설명과 시간 복잡도를 분석하였다. 4절에서는 제안된 휴리스틱 알고리즘의 실험 결과로 픽업 역들과 포착율을 그림과 그래프로 보여주고, 그리고 이 결과가 최적해와 거의 같음을 증명하기 위한 추가 실험 결과를 서술하고 있다. 5절에서는 결론을 정리하고 이 분야의 추후 연구 내용을 설명한다.

## 2. 서울 수도권 지하철 교통 네트워크에서 Pickup Problem

흐름 포착 위치 할당 모델(Hodgson, 1990)이나 픽업 문제의 모델(Zeng, *et al*, 2009)을 서울 수도권 지하철 네트워크에 적용하여 승객의 흐름을 분석하여 픽업 역을 선정하도록 한다. 서울 수도권 교통카드 시스템에서 지하철 승객이 출발역(Origin)에서 도착역(Destination)까지 이동하는 것을 승객 흐름(passenger flow)이라고 부른다. 만약 어떤 서비스 시설물을 설치한 지하철역이 한 승객의 승객 흐름 경로 상에 위치해 있으면, 그 승객 흐름은 그 지하철역에 의해 포착되었다(captured or intercepted)고 여겨진다. 픽업 문제는 주어진 p개의 시설물들에 포착된 승객 흐름의 양을 최대화하도록 서비스 시설물들을 위치하는 지하철역들을 선정하는 것이다. 본 논문에서 고찰하는 픽업 문제는 다음과 같이 제한 조건인 식 (2), (3), (4), (5)를 만족하면서 식 (1)의 목적함수 Z를 최대화하는 것으로 나타낸다.

$$\text{최대화: } Z = \sum_{q \in Q} f_q y_q \tag{1}$$

제한 조건;

$$\sum_{k \in N_q} x_k \geq y_q, \quad \forall q \tag{2}$$

$$\sum_{k \in K} x_k = p \tag{3}$$

$$x_k \in \{0, 1\}, \quad \forall k \in K \tag{4}$$

$$y_q \in \{0, 1\}, \quad \forall q \in Q \tag{5}$$

식 (1)에서 (5)까지 나타난 변수들에 대한 간단한 설명:

$q$ =승객 흐름인 O-D 쌍의 색인 번호(각 쌍에 대한 최단 경로를 포함한다)

$Q$ =모든 O-D 쌍들의 집합

$f_q$ =O-D 쌍 사이의 최단 경로 상의 승객 흐름의 크기

$y_q = \begin{cases} 1, & \text{만약 } f_q \text{가 포착되었으면,} \\ 0, & \text{그렇지 않으면} \end{cases}$

$k$ =잠재적인 시설물의 위치

$K$ =모든 잠재적인 시설물의 위치들의 집합

$x_k = \begin{cases} 1, & \text{만약 한 시설물이 } k \text{에 위치하면,} \\ 0, & \text{그렇지 않으면} \end{cases}$

$N_q$ =승객 흐름  $q$ 를 포착할 수 있는 잠재적인 시설물 위치들의 집합

즉, 승객 흐름  $q$ 의 경로 상에 있는 잠재적인 시설물 위치들의 집합

$p$ =선정되어야 할 시설물들의 개수

위 식에서 나오는 각 변수에 대한 설명을 서울 수도권 지하철 시스템과 지하철 승객들의 출발역과 도착역 사이의 최단 경로 상의 승객 흐름들과 관련하여 설명한다. 한 승객이 출발역(Origin)에서 승차하여 도착역(Destination)에 하차하였다고 하면, 수도권 지하철 시스템에서 두 역 사이의 최단 경로를 계산할 수 있다. 이러한 최단 경로 상의 승객 흐름을 식에서  $q$ 로 표현하고, 모든 승객들의 출발역과 도착역 사이의 승객 흐름들의 집합은  $Q$ 로 나타내고 있다. 한 승객 흐름의 최단 경로  $q$  상으로 이동하는 모든 승객들의 수를 승객 흐름의 크기로  $f_q$ 로 표현하고 있다. 어떤 한 지하철역의 내부에 픽업을 할 수 있는 시설물

을 설치한다고 하면, 이 지하철역의 위치의 색인 번호가  $k$ 에 해당되고, 그리고 시설물들의 잠재적인 위치 집합인  $K$ 는 서울 수도권 전체 지하철역들의 색인 번호 모임이 된다. 한 승객 흐름의 경로인  $q$ 는 출발역과 도착역 사이의 통과역들을 포함하고 있는 데, 경로  $q$ 에 속한 역들의 색인 번호 집합을  $N_q$ 로 나타내고 있다. 마지막 입력 변수는 선정되어야 할 시설물들의 개수인  $p$ 이다.

앞의 입력 변수들을 사용하여 어떤 경로  $q$ 의 포함 여부를 결정하는 변수  $y_q$ 와 어떤 지하철역  $k$ 가 시설물의 위치로 선정되는 변수  $x_k$ 가 결정 변수 역할을 한다. 식 (1)의 목적 함수인  $Z$ 를 최대화하도록 결정 변수인  $y_q$ 와  $x_k$ 를 정하는 것이 픽업 문제에 대한 최선의 해결책을 찾는 것이다. 식 (2)에서  $y_q=1$ 이라면, 경로  $q$ 를 따라가는 흐름은 경로  $p$  상에 최소한 하나의 선정된 픽업 역에 해당되는 시설물이 있어야 된다는 조건을 나타내고, 식 (3)에서 선정되는 픽업 역들의 개수는 정확히  $q$ 개의 지하철역들의 시설물이어야 한다는 조건이다. 제한 조건인 식 (4)와 (5)는  $x_k$ 와  $y_q$ 가 정수 0 또는 1이어야 된다는 조건을 나타내고 있다. 픽업 문제는 NP-hard 문제로 알려져 있고(Hodgson, 1990), 주로 이진 선형 최적화 방법으로 상업용 소프트웨어 패키지로 최적해를 구하고 있다(Zeng, et al,

2009).

### 3. A Greedy Heuristic Algorithm

본 논문에서 고려하는 픽업 문제의 네트워크는 서울 수도권 지하철 시스템 상의 노드들과 링크들로 구성되고, 이 시스템을 이용하는 지하철 승객들의 흐름이 입력으로 주어진다. 노드의 수는 378개로 아주 큰 네트워크에 해당되고 약 500만 명의 지하철 승객의 교통 흐름을 픽업 문제의 입력으로 사용하기에는 너무 큰 양의 데이터가 된다.  $n$ 개의 노드들 중에서  $p$ 개의 픽업 역들을 구하는 모든 조합들을 고려한 후에 최적의 해답을 구하는 방법은 엄청난 컴퓨팅 자원을 소비하고 시간도 아주 오래 걸린다. 본 논문에서는 빠른 시간 내에 거의 최적에 가까운 해답(near optimal solution)을 구할 수 있는 휴리스틱(heuristic) 알고리즘을 모색하여 대규모의 지하철 네트워크에서 승객 흐름을 포착하는 픽업 역들을 찾아내고자 한다.

Non-Multiple Coverage Greedy Heuristic은 단계별로 가장 큰 흐름을 갖는 노드를  $p$ 개 선정하는 방법

```

모든 지하철 승객 트랜잭션을 계산에 포함한다;
p1=0;
while(p1<p) {
    step 1: 계산에 포함된 모든 지하철 승객의 트랜잭션에 대하여 다음을 실행한다;
        step 1.1: 각 승객의 승차와 하차 사이의 최단 이동 경로를 찾는다;
        step 1.2: 각 승객의 이동 경로 중에서 이미 선정된 pickup 역이 포함되어 있으면 계산에서 제외한다;
        step 1.3: pickup 역을 거치지 않는 이동 경로에 포함된 각 역의 통과 승객수를 1씩 증가한다;
    step 2: 가장 큰 통과 승객수를 가지는 노드를 pickup 역으로 선정한다;
    p1++;
}
    
```

그림 1. 추상적인 greedy pickup 알고리즘

이다(Hodgson, 1990). 한 노드가 선정되면, 그 노드를 통해 지나가는 모든 흐름들은 다음 단계의 계산에서 제외된다. 이 방법을 서울 수도권 지하철 시스템에 적용하면 그림 1과 같은 추상적인 알고리즘을 작성할 수 있다:

앞에서 설명한 추상적인 알고리즘은 하나의 픽업역을 선정할 때마다 모든 지하철 승객들의 승차역과 하차역 사이의 최단 경로를 구하고 다시 통과역들의 흐름을 증가하는 과정이 반복되어 좋은 알고리즘이라 볼 수 없다. 이런 문제점을 해결하는 방법으로 그림 2에서 기술한 것과 같이 한 번 모든 승객들의 교통 트랜잭션들을 읽어서 승차-하차 행렬(O-D Matrix)을 만들어 단계별로 픽업역을 결정할 때 이용하도록 한다. 출발역 색인 번호가  $i$ 이고 도착역 색인 번호가  $j$ 인 승객 흐름  $q$ 의 크기를 나타낸  $f_{ij}$ 를 ODMatrix[i][j]로 표현된다.

그림 2의 함수 MakeODMatrix()의 시간 복잡도를 분석해보면  $T(m,n)=O(n^3+m)$ 과 같다. 여기서  $n$ 은 지하철 시스템 상의 지하철역에 해당되는 노드들의 개수이고,  $m$ 은 승객 흐름을 계산하기 위한 지하철 승객 트랜잭션들의 개수를 나타낸다. 첫 번째 항목인  $O(n^3)$ 은 주어진  $n$ 개의 노드들을 갖는 네트워크에서 all-to-all 최단 경로를 구하는 Floyd-Warshall 알고리

즘(문병로 외, 2005)의 복잡도에 해당되고, 두 번째 항목은  $m$ 개의 지하철 승객들의 통행 흐름의 출발역과 도착역을 추출해내고, 해당 ODMatrix의 flow값을 증가시키는 것으로  $O(m)$ 이 된다. 이 두 항목들을 더하게 되면  $T(m,n)$ 의 시간 복잡도를 얻게 된다.

그림 3의 함수인 ComputeNodeFlow()는 픽업역으로 선정되지 않은 것을 의미하는 pickup\_flag가 false인 지하철역들만 통과하는 하나의 OD 경로가 있으면, 그 경로 상의 각 지하철역의 승객 흐름을 그 OD 경로로 지나가는 통과 승객수 만큼 추가한다. 모든 OD 경로들에 대하여 최단 경로 상의 지하철역의 승객 흐름을 합하면, 각 지하철역의 최종 통과 승객수가 결정된다. 이 함수의 시간 복잡도는  $n \times n \times n_{shortest-path}$ 에 비례한다.  $n_{shortest-path}$ 는 OD 경로에 있는 지하철역들의 개수로 항상 전체 지하철역들의 개수인  $n$ 보다 작다. 그러므로 이 함수의 시간 복잡도는  $T(n)=O(n^3)$ 이 된다.

그림 4는 greedy heuristic 알고리즘으로  $p$ 개의 픽업역을 찾는 것을 설명하고 있다. 먼저 서울 수도권 지하철 네트워크에서 지하철 승객들에 대한 OD matrix를 만든다. 그런 후에 픽업역으로 선정되지 않았다는 의미를 갖도록 모든 지하철역들의 pickup\_flag를 false로 둔다. while 문장은  $p$ 개의 픽

```
void MakeODMatrix()
{
    ODMatrix[n][n]를 설정하고, ODMatrix[i][j].flow의 값을 0으로 초기화한다; // n은 지하철역들의 개수
    Floyd-Warshall algorithm으로 모든 지하철역 사이의 최단 경로를 구한다;
    do {
        지하철 승객 트랜잭션 파일에서 한 승객의 트랜잭션을 읽는다;
        그 트랜잭션으로부터 출발역(origin)과 도착역(destination)을 추출해낸다;
        ODMatrix[origin][destination].flow의 값을 1 증가시킨다;
    } while (end of the transaction file);
}
```

그림 2. O-D Matrix를 만드는 함수

```
void ComputeNodeFlow()  
{  
    pickup_flag가 false인 모든 지하철역의 flow를 0으로 둔다;  
  
    for (i = 0; i < n; i++)    // n is the number of subway stations  
        for (j = 0; j < n; j++)  
            if (ODMatrix[i][j].flow > 0) {  
                출발역 i와 도착역 j 사이의 최단 경로 상의 모든 지하철역들의 pickup_flag가 false이면, 최  
                단 경로상의 모든 역의 승객 흐름은 승객 흐름(ODMatrix[i][j].flow) 만큼 증가시킨다;  
            }  
}
```

그림 3. 각 노드의 승객 흐름을 계산하는 함수

```
void GreedyPickupStations(int p)  
{  
    call MakeODMatrix();    // 그림 2에 설명  
    모든 지하철역의 pickup_flag를 false로 둔다;  
    p1 = 0;  
    while (p1 < p) {    // until p pickup stations  
        call ComputeNodeFlow();    // 그림 3에 설명  
        모든 지하철 역 중에서 역의 pickup_flag가 false인 것 중에서 승객 흐름이 최대인 지하철역을 찾아  
        서 해당 지하철역의 pickup_flag를 true로 둔다;  
        포착율(capturing ratio)을 계산한다;  
        p1++;  
    }  
}
```

그림 4. Pickup 역을 찾는 Greedy Heuristic algorithm

업 역을 선정할 때 까지 그 내부의 실행문들을 수행한다. 첫 번째 내부 실행문은 ComputeNodeFlow() 함수를 부름으로 pickup\_flag가 false인 모든 지하철역들의 통과 승객수를 구한다. 그 다음 내부 실행문에서 이 지하철역들 중에서 통과 승객수가 최대인 지

하철역을 픽업 역으로 선정한다. 세 번째 내부 실행문은 선정된 픽업 역이 다음 승객 흐름의 계산에서 포함되지 않도록 pickup\_flag를 true로 둔다. 즉, 어떤 승객의 출발지와 도착지 사이의 최단 경로 상에서 하나라도 pickup\_flag가 true인 지하철역이 있다면,

이 흐름은 다음 픽업 역 선정에 포함되지 않는다. 그림 4의 greedy heuristic 알고리즘의 시간 복잡도는 그림 2와 3의 함수에 대한 시간 복잡도를 고려하면,  $T(m,n)=O(m+pn^3)$ 가 된다. 첫 번째 항목은 OD matrix를 만드는 과정의 시간을 나타내고, 두 번째 항목은 p번의 모든 지하철역의 승객 흐름을 계산하는 것을 표현한다. p개의 최적의 픽업 역들을 찾아내는 방법으로 n개의 지하철역들에서 p개를 뽑아내는 모든 조합들에 대한 승객 흐름을 계산하여 그 중에서 가장 큰 값을 갖는 조합을 찾아내는 brute-force 방법을 생각해 볼 수 있다. 이 방법의 시간복잡도는  $O(m+\binom{n}{p}n^3)$ 이 된다. 제안된 휴리스틱 알고리즘의 시간복잡도와 비교하면 두 번째 항목이  $\binom{n}{p}n^3$ 로 차이가 나고, 이 항목으로 인하여 brute-force 방법으로 해답을 구하면 픽업 문제가 NP-hard 문제임을 알게 해준다. 서울 수도권 지하철 시스템에서와 같은 경우 n은 378이 되고 픽업 역들의 개수로 p를 4로 두면 brute-force 방법에 의한 계산되어야 할 조합들의 개수는  $\binom{378}{4}=837,222,750$ 개나 되어 계산결과를 얻기에는 메모리 문제와 실행 시간의 문제로 인하여 아주 어려운 상황이 된다. 실험 결과에서  $\binom{378}{3}=8,930$ ,

376개의 조합들에 대한 해답을 얻는 데 약 47시간 정도 걸렸다. 그러므로 픽업 역이 4개 이상인 경우인  $p \geq 4$ 에 대해서 p개의 지하철역들을 갖는 모든 조합들을 계산하여 해답을 얻는 방식은 현실적이지 못함을 알 수 있다.

### 4. 실험 결과 및 분석

2007년 기준의 서울 수도권 지하철 시스템에서 승객들이 승차하고 하차하는 트랜잭션들을 입력 데이터로 하여 제안된 greedy heuristic algorithm으로 pickup 역을 찾았다. 실험에 사용된 컴퓨터는 Intel i7 920 CPU, 24GB의 주메모리 시스템, Microsoft Windows 7 64bit 운영체제, 그리고 Microsoft Visual Studio 2008 C++ Programming Language를 사용하였다.

서울 수도권 시스템의 대략적인 구성에 대한 것은 그림 6에 나타나 있다. 전체 지하철역들의 개수는 378개이고, 지하철역들을 이어주는 간선의 개수는 422개이다. 2007년 5월 16일 하루 동안 서울 수도권 승객들이 버스와 지하철을 사용한 결과인 트랜잭션 데이터베이스를 사용하였다. 이 트랜잭션 데이터베

표 1. 선정된 pickup 역의 통과 승객수 및 포착율

순위	포착율(%)	지하철역명	통과승객수	누적승객수	순위	포착율(%)	지하철역명	통과승객수	누적승객수
1	17.9010	신도림	926,376	926,376	11	73.0267	합정	120,925	3,779,133
2	30.1438	교대	633,568	1,559,944	12	75.2779	왕십리	116,501	3,895,634
3	41.5645	동대문운동장	591,019	2,150,963	13	76.9064	태릉입구	84,277	3,979,911
4	47.6863	건대입구	316,804	2,467,767	14	78.5165	모란	83,323	4,063,234
5	53.6429	종로3가	308,255	2,776,022	15	80.1149	신길	82,713	4,145,947
6	58.3210	충신대입구	242,091	3,018,113	16	81.4601	가산디지털단지	69,619	4,215,566
7	61.9244	부평	186,475	3,204,588	17	82.6208	시청	60,064	4,275,630
8	64.9672	창동	157,468	3,362,056	18	83.6779	선릉	54,706	4,330,336
9	67.8388	잠실	148,605	3,510,661	19	84.6899	동묘앞	52,368	4,382,704
10	70.6900	금정	147,547	3,658,208	20	85.6252	서울대입구	48,403	4,431,107

이스에는 10,856,337개의 트랜잭션들을 포함하고 있고, 알고리즘의 입력으로 사용된 정확한 지하철 트랜잭션들의 개수는 5,175,004개이다.

표 1은 greedy heuristic algorithm을 적용하여 상위 20개 까지 픽업 역을 나타내고 각 순위에 따른 누적 통과 승객수를 보여주고 있다. 여기서 포착율(capturing ratio 또는 intercepting ratio)은 누적 통과 승객수를 전체 승객수인 5,175,004를 나눈 퍼센티지 값이다. 그림 5는 픽업 역이 50개 까지의 누적 통과 승객수에 대한 포착율을 보여주고 있다. 50번째의 픽업 역은 “한성대입구”역이고 포착율은 96.3507%이다. 100%의 포착율에 대한 실험 결과는 주어진 입력 상황에서 205번째 픽업 역으로 선정된 “세마”역이고 그 때까지의 누적 통과 승객수는 5,175,004명이다. 실험을 수행한 컴퓨터에서 이 결과를 구하는 데 실행된 시간은 118.2초가 걸렸다.

그림 6은 표 1에서 상위 10개의 픽업 역으로 선정된 지하철역들의 위치를 이해하기 쉽게 서울 메트로 노선도(Seoul Metro, 2007) 상에 검은 색으로 채워진 원으로 표시하였다. 10개 역들의 공통된 특성은 환승

역이고, 1호선과 2호선과 교차하는 9개의 환승역인 “신도림”, “교대”, “동대문운동장”, “건대입구”, “종로3가”, “부평”, “창동”, “잠실”, “금정”역이고, 나머지 한 역은 4호선과 7호선이 만나는 환승역인 “충신대입구”역을 나타내고 있다.

다음에 설명되는 것은 제안된 heuristic 알고리즘으로 찾은 픽업 역이 실제로 최적해(optimal solution)와 같거나 아주 근접되어 있음을 보여주기 위하여 추가적인 여러 실험 결과를 서술하고 있다. 표 2는 첫 번째 픽업 역을 선정하기 전에 초기 지하철역의 통과 승객수를 계산한 것들 중에서 상위 30위 까지 지하철역을 표시하고 있다. 이 표에서 첫 번째 픽업 역으로 “신도림”역이 선정된다. 이 표에서 bold font로 표시된 지하철역명은 표 1에서 상위 9위 까지 선정된 픽업 역이고 괄호 안의 숫자는 선정 순위이다. 그러나 표 2에서는 픽업 선정 순위로 6, 7, 8번째 역에 해당하는 “충신대입구”, “부평”, “창동”역은 나타나지 않고 있다. “교대”역은 두 번째 픽업 역으로 선정되는데, 첫 번째 통과자 수에서 신도림역을 통과하는 승객들의 수를 제외하면 633,568명이 되므로, “신도림”

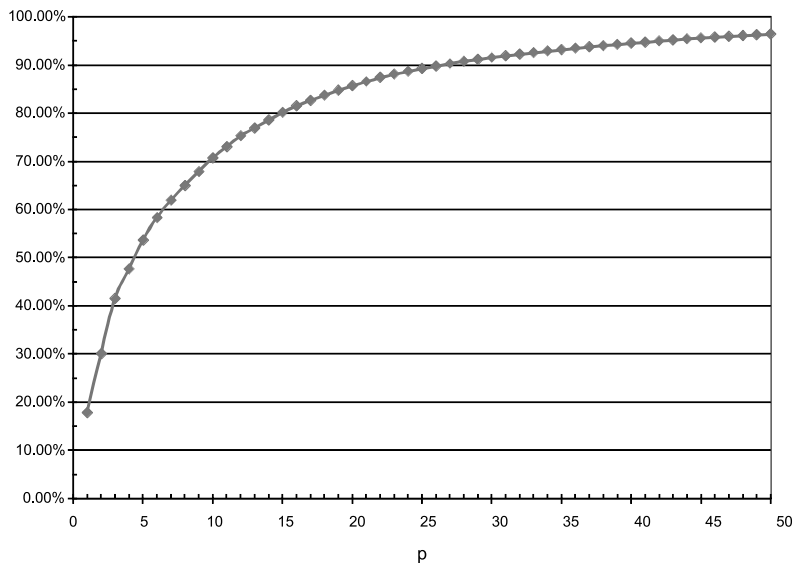


그림 5. Pickup 역의 수에 따른 통과 승객들의 포착율



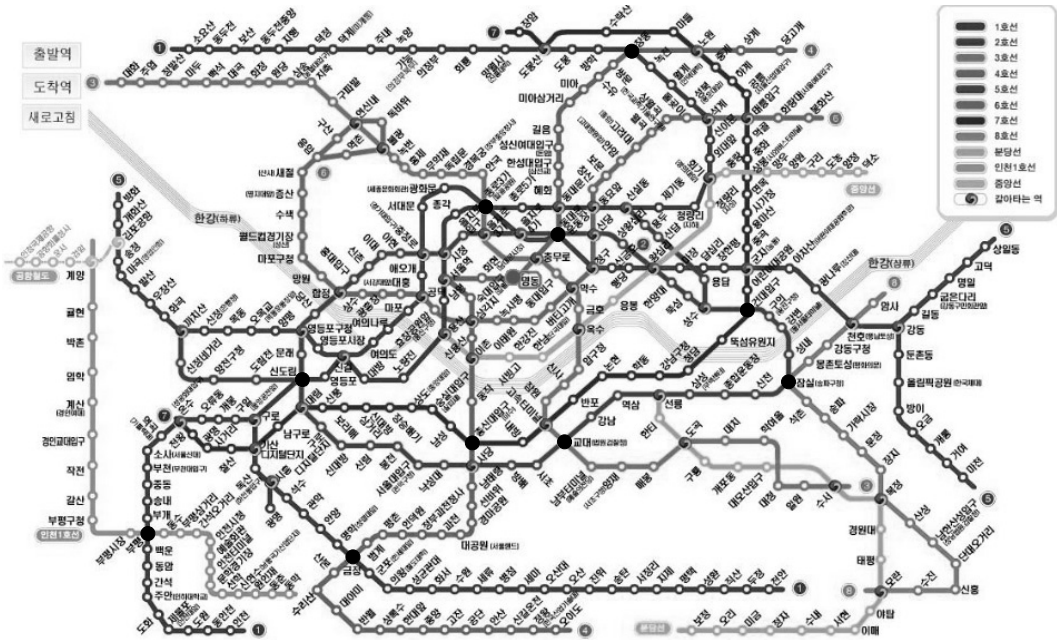


그림 6. 제안된 방법으로 선정된 상위 10개의 pickup 지하철역들

역과 “교대”역을 동시에 통과하는 승객수는 765,579 - 633,568 = 132,011명이 된다. 표 2에서 전체 지하철 승객들이 사당역을 통과하는 승객수는 687,316명으로 표 2에서 3위에 해당되나, 표 1에서 픽업 역으로 첫 번째와 두 번째로 선정된 “신도림”역과 “교대”역으로 인하여 같은 2호선이고 앞의 두 역 사이에 있음으로 하여 픽업 역 선정에서는 40번째로 선정되었다.

표 3은 픽업 문제에서 p=3인 경우의 최적해를 구하는 brute-force 방법의 결과를 보여주고 있다. 지하철역 전체 개수인 378개 중에서 3개를 선택하는  $\binom{378}{3} = 8,930,376$ 개의 조합들 중에서 지하철 승객 흐름을 포착하는 상위 5개의 조합들을 나타낸다. 이 결과는 표 1의 픽업 역의 상위 3개 역과 같음을 알 수 있다. 이 결과를 구하는 데 47시간 정도 걸렸다.

p=4인 경우의 최적해를 구하는 brute-force 방법에서 n 지하철역들에서 4개를 뽑아내는 조합들의 수는

$\binom{378}{4} = 837,227,750$ 개로 프로그램의 메모리 문제와 실행 시간 때문에 그 결과를 쉽게 얻을 수 없다. 표 1의 4번째 순서부터 선정되는 지하철역들의 타당성을 검증하기 위하여 이미 구한 픽업 역들을 제외한 나머지 역들 중에서 통과 승객수가 많은 150개 역들을 선정하여 그 역들에서 3개의 지하철역들의 조합을 구하여 각 조합들의 통과 승객수가 가장 많은 조합 1개를 찾아낸다. 이 경우의 조합들의 개수는  $\binom{150}{3} = 551,300$ 개이다. 이런 방법으로 통과 승객수를 계산하고 그 중에서 가장 많은 승객수를 갖는 지하철역들의 이름은 표 1에서의 지하철역 이름과 같았다.

표 4는 표 2의 초기 지하철 통과 승객수에서 상위 30개 중에서 10개를 선정하는  $\binom{30}{10} = 30,045,015$ 개의 조합들 중에서 가장 많은 승객 흐름을 갖는 지하철역의 조합은 (신도림, 사당, 왕십리, 종로3가, 동대문,

표 2. 초기 지하철역의 통과 승객수(bold font는 9위 까지 선정된 픽업 역)

순위	지하철역명	통과승객수	순위	지하철역명	통과승객수	순위	지하철역명	통과승객수
1	<b>신도림(1)</b>	926,376	11	대림	499,407	21	영등포	449,250
2	<b>교대(2)</b>	765,579	12	시청	494,400	22	낙성대	448,492
3	사당	687,316	13	고속터미널	480,177	23	을지로3가	440,059
4	<b>동대문운동장(3)</b>	638,577	14	방배	477,527	24	서울대입구	438,667
5	왕십리	611,554	15	<b>건대입구(4)</b>	474,619	25	충무로	432,505
6	구로	593,352	16	서초	472,723	26	노량진	431,905
7	<b>종로3가(5)</b>	585,619	17	옥수	468,990	27	서울역	429,030
8	동대문	565,720	18	용산	461,449	28	대방	421,272
9	신길	528,534	19	선릉	457,080	29	신림	414,253
10	강남	520,419	20	역삼	451,373	30	<b>잠실(9)</b>	411,625

표 3.  $\binom{n}{3}=8,930,376$ 개의 조합들 중에 포착율 상위 5 조합들

순위	포착율(%)	통과승객수	지하철역명
1	41.5645	2,150,963	신도림, 동대문운동장, 교대
2	39.7655	2,057,865	신도림, 왕십리, 교대
3	39.6249	2,050,588	동대문, 신도림, 교대
4	39.1320	2,025,085	신도림, 동대문운동장, 사당
5	38.8725	2,011,655	종로3가, 신도림, 교대

표 4.  $\binom{30}{10}=30,045,015$ 개의 조합들 중에서 포착율 상위 5개 조합들

순위	포착율(%)	통과승객수	지하철역명
1	67.84163	3,510,807	신도림, 사당, 왕십리, 종로3가, 동대문, 시청, 고속터미널, 건대입구, 선릉, 잠실
2	67.65585	3,501,193	신도림, 사당, 왕십리, 종로3가, 동대문, 신길, 시청, 고속터미널, 건대입구, 선릉
3	67.43034	3,489,523	신도림, 사당, 왕십리, 종로3가, 동대문, 신길, 시청, 고속터미널, 건대입구, 잠실
4	67.42476	3,489,234	신도림, 사당, 왕십리, 종로3가, 동대문, 시청, 고속터미널, 건대입구, 역삼, 잠실
5	67.42244	3,489,114	신도림, 사당, 왕십리, 구로, 종로3가, 동대문, 시청, 고속터미널, 건대입구, 선릉

시청, 고속터미널, 건대입구, 선릉, 잠실)이고, 이 조합의 지하철역들을 통과하는 승객 흐름은 3,510,807 명으로 포착율은 67.8416%가 된다. 이 포착율은 표 1에서의 값인 70.69%보다 작다. 그 이유는 표 2에서 상위 30개 역들 중에서 표 1에서의 10개 픽업 역들에 속하는 개수는 6개뿐이다. 조합의 구성이 조금 다르게 되면 결과가 아주 많이 틀려짐을 알 수 있고, 제안

된 휴리스틱 알고리즘이 더 좋은 포착율을 가짐을 알 수 있다. 이 결과를 구하는 데 7일 정도 시간이 걸렸다.

표 5는 표 2에서 초기 지하철역의 통과 승객수들이 많은 상위 역들의 개수를 50개로 확장한 역들 중에서 5개 픽업 역들을 뽑아내는  $\binom{50}{5}=2,118,760$ 개의 조합

표 5.  $\binom{50}{5}=8,930,376$ 개의 조합들 중에 포착율 상위 5 조합들

순위	포착율(%)	통과승객수	지하철역명
1	53.6429	2776022	신도림, 교대, 동대문운동장, 종로3가, 건대입구
2	53.6112	2774383	신도림, 교대, 왕십리, 동대문, 충신대입구
3	53.2647	2756449	신도림, 사당, 왕십리, 동대문, 고속터미널
4	53.2480	2755587	신도림, 교대, 왕십리, 동대문, 건대입구
5	53.0159	2743577	신도림, 교대, 동대문운동장, 종로3가, 충신대입구

들 중에서 포착율이 높은 5개의 픽업 역들의 조합을 보여준다. 이 결과를 구하는 데 약 12시간 정도 걸렸다. 표 5의 순위 1의 조합에 나온 역들은 표 1에서 선정된 상위 5개 역의 이름들과 같음을 보여 주어서 제안된 휴리스틱 알고리즘이 정확한 픽업 역들을 선정하고 있음을 검증해주는 결과 중의 하나이다.

## 5. 결론

교통 네트워크에서 흐름 포착 위치 할당 모델(flow capturing location allocation model)의 최적해(optimal solution)를 구하는 것은 NP-hard 문제로 알려져 있고, 기존의 연구도 제한된 범위에서 상업용 최적화 소프트웨어 패키지(예, ILOG CPLEX 12.2)를 사용하여 선형 및 정수 최적화 모델을 이용하여 최적해를 구하고 있다. 이 모델을 픽업(pickup) 문제로 변환하여 서울 수도권 지하철 시스템에다 적용하면, 네트워크의 크기와 500만 개에 이르는 승객 흐름의 양으로 인하여 상업용 소프트웨어를 이용하기에도 어려운 실정이다.

본 논문에서는 이를 해결하는 방법으로 greedy heuristic algorithm을 제안하여 픽업 지하철역들을 빠른 시간 내에 계산하도록 하였고, 결과가 거의 최적해와 같음을 여러 실험을 통하여 보여주었다. 논문의 실험 결과를 적용하여 신선 농수산물을 픽업하는 서비스 지하철역을 결정하는 데 이용될 수 있고, 또

한 제안된 방법은 전기 자동차의 급속 충전소의 위치 선정 등에 이용될 수 있다. 추후 연구로는 상업용 최적화 소프트웨어를 활용하여 최적해를 구하여 제안된 휴리스틱 알고리즘의 결과와 비교 검토하는 것이다.

## 참고문헌

- 박종수 · 이금숙, 2010, "서울 수도권 지하철 교통망에서 승객 흐름의 분석," 한국정보과학회논문지: 컴퓨팅의 실제 및 레터 16(3), pp.316-323.
- 문병로 · 심규석 · 이충세 역, 2005, *Introduction to Algorithms, 2nd Ed.*, 서울, 한빛미디어, pp.680-687.
- Hodgson, M. J., 1990, "A Flow-Capturing Location-Allocation Model," *Geographical Analysis* 22(3), pp.270-279.
- Zeng, W., Hodgson, M. J. and Castillo, I., 2009, "The Pickup Problem: Consumers' Locational Preferences in Flow Interception," *Geographical Analysis* 41, pp.107-126.
- Linesa, L., Kubyb, M., Schultzc, R., Clancyb, J., and Xiec, Z., 2008, "A rental car strategy for commercialization of hydrogen in Florida," *international journal of hydrogen energy* 33, pp.5312-5325.
- Homera, M. W. and Groves, S., 2007, "Network flow-based strategies for identifying rail park-and-ride facility locations," *Socio-Economic Planning Sciences* 41, pp.255-268.

- Kuby, M. and Li, S., 2005, "The flow-refueling location problem for alternative-fuel vehicles," *Socio-Economic Planning Sciences* 39, pp.125-145.
- ReVelle, C. S. and Eiselt, H. A., 2005, "Location analysis: A synthesis and survey," *European Journal of Operational Research* 165, pp.1-19.
- Hodgson, M. J., Rosing, K. E., Leontien, A., and Storrier, G., 1996, "Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada," *European Journal of Operational Research* 90, pp.427-443.
- Lee, K., Jung, W. -S. Park, J. S., and Choi, M.Y., 2008, "Statistical analysis of the Metropolitan Seoul Subway System: Network structure and passenger flows," *Physica A: Statistical Mechanics and its Applications* 387(24), pp.6231-6234.
- Lee, K., Goh, S., Park, J. S., Jung, W. -S., and Choi, M. Y., 2011, "Master equation approach to the intra-urban passenger flow and application to the Metropolitan Seoul Subway system," *Journal of Physics A: Mathematical and Theoretical* 44, pp.115007-115020.
- Seoul Metro, 2007, <http://www.seoulmetro.co.kr/>
- 교신: 이금숙, 서울특별시 성북구 동선동 3가 249-1, 사회과학대학 지리학과, 전화: 02-920-7138, 이메일: kslee@sungshin.ac.kr
- Correspondence: Keumsook Lee, 249-1 Dongseon-dong 3-ga, Seongbuk-gu, Seoul 136-742, Korea, Tel: +82-2-920-7138, e-mail: kslee@sungshin.ac.kr

최초투고일 2011년 6월 3일  
최종접수일 2011년 6월 20일

## **Greedy Heuristic Algorithm for the Optimal Location Allocation of Pickup Points: Application to the Metropolitan Seoul Subway System**

Jong Soo Park\* · Keumsook Lee\*\*

**Abstract** : Some subway passengers may want to have their fresh vegetables purchased through internet at a service facility within the subway station of the Metropolitan Seoul subway system on the way to home, which raises further questions about which stations are chosen to locate service facilities and how many passengers can use the facilities. This problem is well known as the pickup problem, and it can be solved on a traffic network with traffic flows which should be identified from origin stations to destination stations. Since flows of the subway passengers can be found from the smart card transaction database of the Metropolitan Seoul smart card system, the pickup problem in the Metropolitan Seoul subway system is to select subway stations for the service facilities such that captured passenger flows are maximized.

In this paper, we have formulated a model of the pickup problem on the Metropolitan Seoul subway system with subway passenger flows, and have proposed a fast heuristic algorithm to select pickup stations which can capture the most passenger flows in each step from an origin-destination matrix which represents the passenger flows. We have applied the heuristic algorithm to select the pickup stations from a large volume of traffic network, the Metropolitan Seoul subway system, with about 400 subway stations and five millions passenger transactions daily. We have obtained not only the experimental results in fast response time, but also displayed the top 10 pickup stations in a subway guide map. In addition, we have shown that the resulting solution is nearly optimal by a few more supplementary experiments.

**Keywords** : passenger flow, traffic network, pickup problem, flow capturing location allocation model, heuristic algorithm, optimization problem, Metropolitan Seoul Subway system

---

\* Professor, School of Information Technology, Sungshin Women's University

\*\* Professor, Department of Geography, Sungshin Women's University