

# 우편집중국간 우편물 운송계획 문제의 타부 탐색 알고리즘

최지영\*, 송영효\*\*, 강성열\*\*\*

## A Tabu Search Algorithm for the Postal Transportation Planning Problem

Ji young Choi, Young hyo Song, Sung yeol Kang

### Abstract

This paper considers a postal transportation planning problem in the transportation network of the form of hub and spoke. Given mail sorting centers and an exchange center, available vehicles and amount of mails to be transported between mail sorting centers, postal transportation planning is to make a transportation plan without violating various restrictions. The objective is to minimize the total transportation cost.

To solve the problem, a tabu search algorithm is proposed. The algorithm is composed of a route construction procedure and a route improvement procedure to improve a solution obtained by the route construction procedure using a tabu search. The tabu search uses the best-admissible strategy, BA, and the first-best-admissible strategy, FBA. The algorithm was tested on problems consisting of 11, 16 and 21 mail sorting centers including one exchange center. Solutions of the problems consisting of 11 mail sorting centers including one exchange center were compared with optimal solutions. On average, solutions using BA strategy were within 0.287% of the optimum and solutions using FBA strategy were within 0.508% of the optimum. Computational results show that the proposed algorithm can solve practically sized problems within a reasonable time and the quality of the solution is very good.

Key Words : Postal Transportation Planning, Tabu Search

\* 한국전자통신연구원 우정기술연구센터

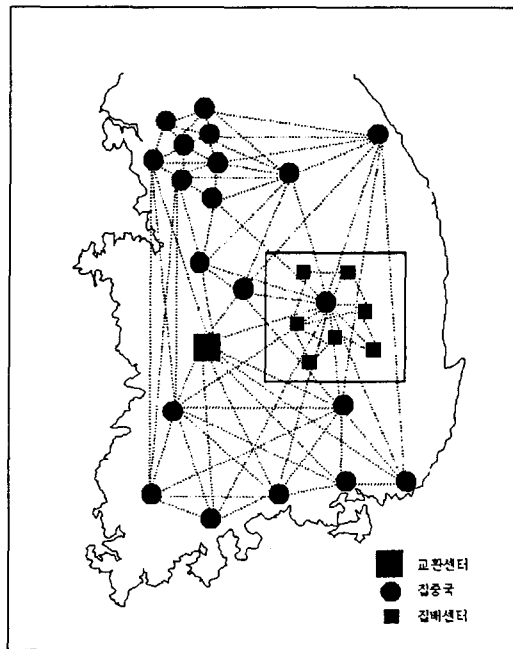
\*\* 홍익대학교 경영정보학과

\*\*\* 홍익대학교 경영정보학과

## 1. 서론

우체국에서는 현재 철도와 직체결국(108개국)을 중심으로 하는 점 대 점(point-to-point) 형태의 운송망을 우편물 교환센터를 중심으로 우편물을 상호 교환

하는 허브와 스포크(hub and spoke) 형태의 운송망으로 개편할 계획이다. 그리고 우편 집중국(22개국)을 건설하여 기존의 우체국간 운송망에서 집중국간 운송망으로 전환할 계획이다. (그림 1)은 목표로 하고 있는 운송망의 형태를 나타낸다.



(그림 1). 허브와 스포크 형태의 운송망.

(그림 1)에서 보는 바와 같이 허브와 스포크 형태의 운송망에는 하나의 교환센터를 중심으로 다수의 집중국이 존재한다. 또한 하나의 집중국은 관내에 다수의 집배센터를 포함하고 있으며, 하나의 집배센터는 다수의 고객을 포함하고 있다. 우편물량의 흐름을 보면 우선 고객들의 우편물은 관할 집배센터에 의해 수집되고 이 우편물들은 다시 관할 집중국에 의해 수집된다. 집배센터로부터의 수집이 끝난 후 각 집중국에서는 수집된 우편물을 목적지가 속해 있는 집중국별로 구분해 교환센터의 교환시간에 맞추어 교환센터로 보내며, 교환센터에서는

각 집중국으로부터 온 물량을 목적 집중국별로 교환하는 기능을 수행한다. 교환된 물량은 다시 각 집중국으로 배달되고 집중국에서는 물량을 구분해 관내 집배센터로 배달한다. 집배센터에서는 우편물을 고객에게 배달함으로써 우편물은 최종 목적지에 도착하게 된다.

위와 같은 우편물량의 흐름에 따라 새로운 운송망에서의 운송계획은 크게 세 가지로 나누어진다. 첫째는 하나의 교환센터와 다수의 집중국을 대상으로 하는 전국 운송계획이며, 둘째는 하나의 집중국과 관내 집배센터를 대상으로 하는 지역 운송계

획이다. 셋째는 하나의 집배센터와 관할 구역 내 고객을 대상으로 하는 순로계획이다. 본 연구는 이 중에서 전국 운송계획 수립을 다루고자 한다. 즉 관내 집배센터로부터 수집된 물량을 각 집중국에서 교환센터로 운송하고 교환센터에서 교환된 물량을 다시 각 집중국으로 운송하는 운송경로를 설정하고 설정된 운송경로에 차량을 할당하는 운송계획을 수립하고자 한다.

교환센터와 집중국을 포함한 운송국, 운송국 사이의 운송로, 운송수단, 운송물량 등이 주어진 경우, 전국 운송계획이란 차량의 용량 제한이나 운송국의 시간제한을 만족하며 모든 운송물량을 출발지에서 도착지까지 운송하는 운송경로를 설정하는 것을 의미한다. 이 때 목적함수는 각 경로들의 운송비용을 합한 총 운송비용을 최소화하는 것이다.

전국 운송계획에서 운송계획의 실제 대상이 되는 우편물량은 집중국과 집중국간에 존재한다. 우편물량은 출발 집중국에서 출발해 교환센터를 거쳐 도착 집중국으로 운송되거나, 집중국간 물량이 많은 경우 운송비용 절감을 위해 교환센터를 거치지 않고 출발 집중국에서 도착 집중국으로 직접 운송된다. 또한 각 집중국에서 교환센터로 가는 물량과 교환센터에서 각 집중국으로 오는 물량이 많은 경우 차량의 용량제한 때문에 다른 집중국을 경유하는 것이 불가능하므로 집중국과 교환센터 간에 직접 운송된다. 즉 집중국간 직접 운송이나 집중국과 교환센터간 직접 운송에 의해 운송비용 절감과 동시에 물량을 사전에 처리할 수가 있다.

본 연구에서는 사전에 결정이 가능한 집중국간 직접 운송이나 집중국과 교환센터간 직접 운송에 의해 물량을 우선 처리하고, 남은 물량을 대상으로 운송계획을 수

립하고자 한다. 남은 물량을 대상으로 하는 운송계획은 기존의 시간제한 하에서 backhaul이 있는 차량경로문제(Vehicle Routing Problem with Backhauls and Time Windows, VRPBTW)와 유사하며, VRPBTW에 대해서는 2장에서 자세히 설명한다.

운송계획 문제를 해결하기 위한 방법으로 본 논문에서는 타부 탐색(tabu search) 알고리즘을 제시한다. 알고리즘은 가능 해를 발생시키는 단계와 그 해를 개선하는 단계로 이루어진다. 해의 발생 단계에서는 insertion 휴리스틱[Solomon, 1987]을 변형하여 적용해 가능 해를 생성하고, 해의 개선 단계에서는 해의 발생 단계에 의해 생성된 가능 해를 타부 탐색을 이용해 개선하여 국소 해(local optimum)에 머물지 않고 최적 해에 가까운 해를 찾을 수 있도록 하였다.

2장에서는 본 연구에서 다루고 있는 우편물 전국 운송계획에 대한 문제 설명 및 기존 연구에 대해 설명한다. 3장에서는 우편물 운송계획 문제의 수리 모형을 제시하고, 4장에서는 타부 탐색 알고리즘을 제시한다. 5장에서는 구현된 알고리즘을 적용한 결과에 제시하고, 마지막으로 6장에서는 결론을 제시한다.

## 2. 문제 정의와 기존 연구

### 2.1 문제 정의

운송계획 수립에 있어 대상이 되는 운송국은 하나의 교환센터와 다수의 집중국이다. 운송국 집합을  $V = \{0, 1, 2, \dots, n\}$ 로 나타낼 경우, 교환센터(노드 0)는 각 집중국으로부터 온 물량을 교환하는 기능을 담

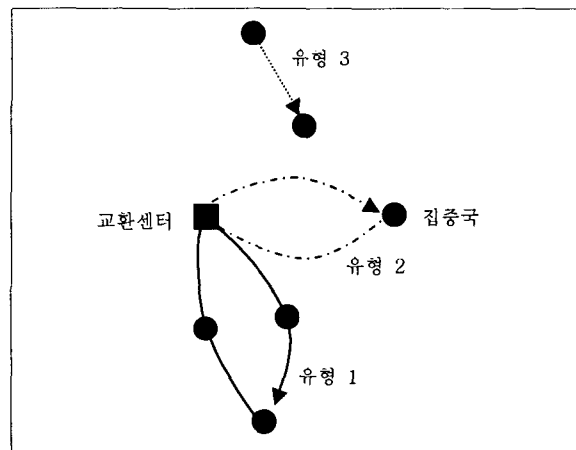
당하며, 집중국(노드  $i, i \in V \setminus \{0\}$ )은 관내 집배센터로부터 수집된 물량을 구분해 교환센터로 보내거나 교환센터로부터 온 물량을 구분해 관내 집배센터로 배달하는 기능을 수행한다. 운송계획 수립이란 이 운송국들 간에 운송경로를 설정하는 것을 의미한다.

집중국과 집중국간 또는 집중국과 교환센터간 운송로는 기본적으로 모두 존재하며 방향성이 없다. 운송로 집합을  $E$ 라 할 때 운송로의 거리  $l_{ij}((i, j) \in E)$ 는 유클리디언 거리(Euclidean distance)이며, 운송로를 운행하는데 있어 소요시간은 거리를 운송수단인 트럭의 속도로 나눈 값을 사용한다.

운송국간 물량을 운송하는 수단으로는 철도와 트럭을 고려한다. 철도의 경우 여러 구간이 존재하며 그 구간에 해당하는 집중국간 물량에 한해 운송이 가능하다. 트럭은 용량을 기준으로 서로 다른 유형이 존재한

다. 운송물량을 팔레트나 트롤리 등의 용기에 담아 트럭에 싣게 되는데 한 트럭에 두 종류 이상의 용기를 적재하는 경우 트럭의 용량과 용기의 부피를 고려한 적재유형 테이블을 기준으로 용기를 적재해야 한다(트럭유형별 용기의 적재유형 테이블은 [최지영, 2001]을 참고).

운송의 대상이 되는 운송물량은 집중국 간에 존재한다. 운송물량은 통상과 소포로 구분되며 이 때 통상은 일반 우편물을 의미한다. 물량을 담는 운송용기는 우편물 유형에 따라 다르다. 통상의 경우 통 단위의 운송물량을 행낭에 담은 후 팔레트에 담아 운송하거나 상자에 담은 후 트롤리에 담아 운송하며, 소포의 경우 행낭에 담은 후 팔레트에 담아 운송한다. 위에서 언급한 바와 같이 철도나 트럭에는 팔레트, 트롤리 등의 운송용기만이 적재가 가능하다.



(그림 2). 운송경로 유형.

운송계획 수립에 있어 고려하는 운송경로 유형은 (그림 2)와 같다. 운송경로란 한 대의 트럭이 물량을 운송하기 위해 경유하는 운송국들의 순서를 의미한다. 1장에서 언급한 바와 같이 모든 물량은 출발 집중국에서 출발해 교환센터를 거쳐 도착 집중국

국으로 운송된다. 이 때 유형 2와 같이 출발 집중국에서 교환센터까지 물량을 운송한 트럭은 교환센터에서 교환이 끝난 후 출발 집중국으로 가는 운송물량을 싣고 출발 집중국으로 돌아온다. 유형 1과 같이 출발 집중국에서 교환센터로 가는 상행경로

나 교환센터에서 집중국으로 돌아오는 하행경로에서 다른 집중국을 경유하는 것이 가능하다. 유형 1이나 유형 2와 같이 출발 집중국과 도착 집중국이 같고 교환센터를 경유하는 경로를 교환센터 경유경로라고 한다. 집중국간 물량이 많은 경우에는 운송비용 절감을 위해 유형 3과 같이 교환센터를 거치지 않고 출발 집중국에서 도착 집중국으로 직접 운송이 가능하며 이와 같은 경로를 직접경로라 한다. 각 집중국에서 교환센터로 가는 물량과 교환센터에서 각 집중국으로 오는 물량이 많은 경우 차량의 용량제한 때문에 다른 집중국을 경유하는 것이 불가능하므로 유형 2와 같이 집중국과 교환센터 간에 물량을 직접 운송한다. 교환센터 경유경로 중 유형 2와 같은 경로를 교환센터 직접경로라 한다.

유형 1과 같은 교환센터 경유경로의 경우 상행경로에서는 상역, 즉 발송만을 허용하며 하행경로에서는 하역, 즉 도착만을 허용한다. 상행경로에서 해당 경유국 사이의 우편물은 상하역이 가능할 수 있다. 하지만 이를 실행하기 위해서는 각 집중국에서의 구분 작업 이외에도 차량 상역시 적재순서 고려 등으로 운송작업이 복잡하게 되므로 고려하지 않는다. 상행경로와 하행경로상의 경유지에서 작업시간은 상하역하는 운송물량에 비례하여 결정된다.

위와 같은 기본적인 정보가 주어졌을 때, 운송계획을 통해 얻고자 하는 정보는 운송경로 설정 및 차량할당 대안으로서 어떤 유형의 차량이 어떤 경로를 통해 언제 얼마만큼의 물량을 운반하고, 이 때 각 경로별 운송비용 및 총 운송비용이 얼마인가 하는 것이다. 운송계획의 목적은 총 운송비용을 최소화하는 것으로 총 운송비용은 철도 운송비용과 트럭 운송비용의 합으로 계산되어진다. 철도 운송비용은 구간에 해당

하는 운송비용을 이용한다. 트럭 운송비용은 각 경로에 할당된 트럭과 관련된 비용으로, 트럭 1대를 추가로 사용하는데 들어가는 고정비용과 이를 운행하는데 드는 변동비용의 합으로 표시된다. 변동비용은 트럭유형과 운송거리, 탑승인원에 따른 운송비용 테이블에 의해 계산된다. 탑승인원으로는 1인과 2인을 고려한다(자세한 운송비용 테이블 자료는 [최지영, 2001]을 참고). 직접경로의 경우 집중국간 거리를 이용해 운송비용을 계산하며, 교환센터 경유경로의 경우 상행경로상의 거리를 고려한 운송비용과 하행경로상의 거리를 고려한 운송비용의 합으로 계산한다.

운송계획 수립에 있어 고려하는 제약조건은 다음과 같다. 우선 운송계획을 통해 모든 운송물량이 처리되어야 한다. 즉 집중국간에 존재하는 모든 운송물량이 출발 집중국에서 출발해 도착 집중국에 도착해야 한다. 다음으로 철도나 트럭에 의해 물량을 운송할 때 적재유형 테이블을 기준으로 운송이 이루어져야 한다. 즉 철도나 트럭의 용량 제한을 만족해야 하며, 교환센터 경유경로의 경우 상행경로와 하행경로에서 각각 트럭의 용량 제한을 만족해야 한다. 마지막으로 운송국들의 시간제한을 만족해야 한다. 시간제한이란 교환센터의 경우 모든 우편물량의 교환이 가능하도록 각 집중국으로부터 우편물량이 도착해야 하는 도착한계시각과 모든 교환이 완료된 후 각 집중국으로 출발이 가능한 출발가능시각을 의미하며, 집중국의 경우 집배센터로부터 수집이 끝나 교환센터로의 발송이 가능한 출발가능시각과 집배센터의 배달 시간에 맞도록 우편물량이 교환센터에서 도착해야 하는 도착한계시각을 의미한다. 따라서 직접경로의 경우 출발 집중국에서 출발가능시각 이후에 출발해 도착 집중국에 도착한

계시각 이전에 도착해야 하며, 교환센터 경유경로의 경우 상행경로상의 집중국에서 출발가능시각 이후에 출발해 교환센터에 도착한계시각 이전에 도착하고 교환센터에서 출발가능시각 이후에 출발해 하행경로상의 집중국의 도착한계시각 이전에 도착해야 한다.

## 2.2 가정

운송계획 수립에 있어 기본적인 가정사항에는 두 가지가 있다. 우선 교환센터와 각 집중국간 운송로의 소요시간에 관한 가정으로 식 (1), 식 (2)와 같다.

$$t_{0i} \leq d_0 - s_i, \forall i \in \mathcal{V} \setminus \{0\} \quad (1)$$

$$t_{0i} \leq d_i - s_0, \forall i \in \mathcal{V} \setminus \{0\} \quad (2)$$

식 (1)은 교환센터와 각 집중국간 운송로의 소요시간  $t_{0i}$ 는 교환센터의 도착한계시각  $d_0$ 와 집중국의 출발가능시각  $s_i (i \in \mathcal{V} \setminus \{0\})$ 의 차보다 작거나 같고, 식 (2)는  $t_{0i}$ 는 집중국의 도착한계시각  $d_i (i \in \mathcal{V} \setminus \{0\})$ 와 교환센터의 출발가능시각  $s_0$ 의 차보다 작거나 같다는 것을 의미한다. 각 집중국의 물량은 기본적으로 교환센터로 운송되고 다시 각 도착 집중국으로 운송되어야 하므로 위의 가정은 문제를 가능(feasible)하게 하기 위해 반드시 필요한 가정이다.

다음으로 운송계획을 수립할 때 운송물량의 적재에 있어 비분할 선적을 가정한다. 비분할 선적이란 발송해야 하는 물량을 하나의 운송경로를 통해서 한 대의 운송수단으로 운송하는 방안이다. 즉 이 방안은 운송물량이 고려대상이 되고 있는 운송수단의 최대용량보다 항상 작거나 같다고 하는

가정을 내재하고 있다. 만약 물량이 많아 한 대의 운송수단으로 운송할 수 없는 경우에는 물량을 미리 분할하여 각각의 분할된 물량에 대해서 비분할 선적을 적용한다.

## 2.3 사전처리 작업

사전처리 작업 단계는 직접경로 및 교환센터 직접경로를 이용해 물량을 미리 처리하고, 남은 물량이 많은 경우 비분할 선적을 적용할 수 있도록 물량을 미리 분할하는 단계이다.

### 2.3.1 직접경로 설정

특정 집중국간의 운송물량이 많은 경우에는 교환센터의 능력과 운송비용 절감을 위해서 해당 집중국간의 직접 경로를 개설하고 목적 집중국으로 직행하는 것이 가능하다. 이 때 직접 경로란 집중국간 운송물량이 트럭의 적재비율 ( $\beta_q, \forall q \in Q$ ) 이상인 경우 한 대의 트럭으로 운송이 가능할 때까지 해당 집중국간에 직접경로를 배정하여 물량을 처리하는 것을 의미한다. 이 때 돌아오는 경로는 고려하지 않으며 제약조건을 만족시키는 경우에 한해 직접경로를 개설한다.

본 문제의 경우 다수의 트럭 유형이 존재하므로 직접경로에 할당할 트럭 유형을 결정해야 한다. 본 연구에서는 운송비용을 최소화하기 위해 트럭 유형별로 운송비용을 계산해서 가장 비용이 적게 드는 트럭 유형을 선택하도록 하였다. 트럭 유형별 운송비용  $C_q$ 는 직접경로로 배정된 트럭의 운송비용  $C_q^1$ 과 잔여물량의 운송비용  $C_q^2$ 의 합으로 계산된다. 직접경로로 배정된 트럭의 운송비용의 경우 할당된 트럭대수에 트럭의 운송비용을 곱해 구할 수 있으나 잔여물량의 운송비용은 이후 알고리즘 단계

에서 이루어지므로 그 값을 미리 알 수가 없다. 그러므로 최종 운송용기별 단위 운송비용을 추정해 잔여물량에 곱하여 그 비용을 계산하였다. 직접경로에 할당된 운송물량의 경우 출발 집중국에서 도착 집중국으로 직접 운송되므로 해당 집중국간 거리만 고려하면 되지만, 잔여물량의 경우 교환센터를 경유해 도착 집중국으로 운송되므로 출발 집중국에서 교환센터까지의 거리와 교환센터에서 도착 집중국까지의 거리의 합을 고려해야 한다. 따라서 최종 운송용기별 단위 운송비용 추정시 이 거리를 고려하여, 잔여물량을 운송할 수 있는 트럭 유형들의 최종용기별 단위 운송비용을 계산하여 평균을 취하였다.

예를 들어 출발 집중국이  $i \in V \setminus \{0\}$ , 도착 집중국이  $j \in V \setminus \{0\}$ , 트럭유형이  $q \in Q$ , 트럭의 적재용량이  $\lambda_q$ 인 경우 운송비용  $C_q$ 는 식 (3)과 같다.  $n$ 은 할당된 트럭대수를 의미하고,  $\alpha$ 는 잔여물량을 의미한다.

$$C_q = C_q^1 + C_q^2 \quad (3)$$

$$C_q^1 = n \times F(q, l_{ij}, k) \quad (4)$$

$$C_q^2 = \alpha \times \left\{ \sum_{q \in Q} F(q, l_{ij}, k) / \lambda(q) \right\} / |Q| \quad (5)$$

### 2.3.2 교환센터 직접경로 설정

직접경로를 설정하고 난 후의 집중국간 운송물량  $r_{ij}(i, j \in V \setminus \{0\})$ 는 모두 트럭 한 대분 미만으로 이 물량은 모두 교환센터를 경유해 도착 집중국으로 운송된다. 따라서 각 집중국별로 남은 운송물량을 더해 하나의 집중국에서 교환센터로 가는 발생물량과 교환센터에서 집중국으로 오는 도착물량을 계산한다. 집중국  $i \in V \setminus \{0\}$ 의 경우 발생물량  $p_i$ 는  $\sum_{j \in V \setminus \{0\}} r_{ij}$ 와 같고 도착물량  $d_i$ 는  $\sum_{j \in V \setminus \{0\}} r_{ji}$ 와 같다.

집중국의 발생물량과 도착물량이 트럭의 적재비율 이상인 경우 한 대의 트럭으로 운송이 가능하며, 이 경우 트럭 용량제한 때문에 다른 집중국을 경유하는 것이 불가능하므로 집중국과 교환센터간 교환센터 직접경로의 개설이 가능하다. 따라서 한 대의 트럭으로 운송이 가능할 때까지 교환센터 직접경로를 개설하여 물량을 처리한다.

직접경로의 경우와 마찬가지로 운송비용을 최소화하기 위해 어떤 트럭유형을 교환센터 직접경로에 할당할지 결정해야 한다. 트럭 유형을 결정하기 위해 트럭 유형별로 운송비용을 계산해서 가장 비용이 적게 드는 트럭 유형을 선택하도록 하였다. 트럭 유형별 운송비용은 교환센터 직접경로로 배정된 트럭의 운송비용과 잔여물량의 운송비용 합으로 계산된다. 교환센터 경유경로로 배정된 트럭의 운송비용은 할당된 트럭대수에 트럭의 운송비용을 곱해 구하며, 최종 운송용기별 단위 운송비용은 집중국과 교환센터 거리의 두 배에 해당하는 거리를 고려하여 잔여물량을 운송할 수 있는 트럭 유형들의 최종용기별 단위 운송비용을 계산해 평균을 취하였다. 집중국  $i \in V \setminus \{0\}$ 의 트럭  $q \in Q$ 에 대한 운송비용  $C_q$ 는 식 (6)과 같다.

$$C_q = C_q^1 + C_q^2 \quad (6)$$

$$C_q^1 = n \times F(q, 2 \times l_{ij}, k) \quad (7)$$

$$C_q^2 = \alpha \times \left\{ \sum_{q \in Q} F(q, 2 \times l_{ij}, k) / \lambda(q) \right\} / |Q| \quad (8)$$

### 2.3.3 물량 분할

직접경로와 교환센터 직접경로를 설정한 후 각 집중국의 발생물량과 도착물량은 각각 트럭 한 대 미만이거나, 발생물량과 도착물량이 균등하지 않을 경우 한 쪽 물

량만 트럭 한 대 이상인 경우가 있을 수 있다. 물량이 트럭 한 대 이상인 경우 운송 계획을 수립하기 위해서는 비분할 선적의 가정에 맞게 물량을 미리 분할해야 한다.

직접경로나 교환센터 직접경로의 경우와 마찬가지로 물량 분할시 분할 기준이 되는 트럭유형을 결정하기 위해 트럭유형별로 운송비용을 계산해서 가장 비용이 적게 드는 트럭유형을 선택하도록 하였다. 트럭유형별 운송비용은 사전 처리 작업을 하고 남은 물량을 트럭에 할당했을 때의 트럭대수에 트럭 한 대의 운송비용을 곱해 계산하였다. 집중국  $i \in V \setminus \{0\}$ 의 트럭  $q \in Q$ 에 대한 운송비용  $C_q$ 는 식 (9)와 같다.

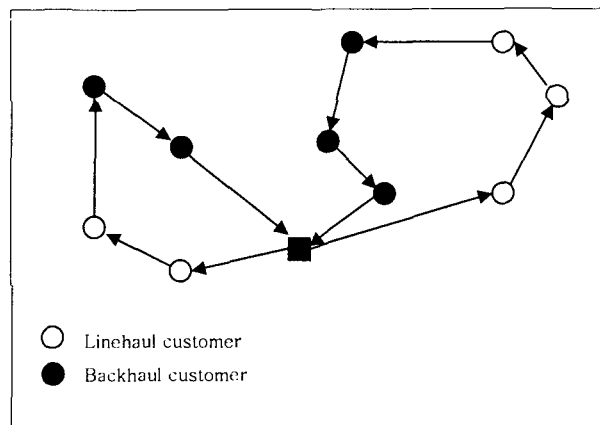
$$C_q = n \times F(q, 2 \times l_0, k) \quad (9)$$

분할 기준이 되는 트럭유형을 결정해 물량을 분할한 후 그 수에 맞게 집중국을

생성하고, 다시 하나의 집중국  $i$ 를 출발 집중국  $i \in V_1$ 와 도착 집중국  $i' \in V_2$ 으로 따로 생성해  $i$ 에는 발생물량을 할당하고  $i'$ 에는 도착물량을 할당한다. 운송계획을 수립하는 알고리즘 단계에서는 가능한 경로로  $i$ 에서 출발해 교환센터를 지나  $i'$ 으로 돌아오는 경로를 고려한다.

## 2.4 기존 연구

본 연구에서는 직접경로나 교환센터 직접경로에 의해 물량을 우선 처리하고, 남은 물량의 경우 비분할 선적의 가정에 맞게 물량을 분할하는 사전처리 작업을 거친 상태에서 운송계획 수립을 다루고자 한다. 1장에서 언급했듯이 사전처리 작업을 거친 후의 운송계획 문제는 VRPBTW와 유사하다.



(그림 3). backhaul이 있는 차량경로문제.

Backhaul이 있는 차량경로문제(Vehicle Routing Problem with Backhauls, VRPB)는 수집 또는 배달 중 하나만 수행하는 일반적인 차량경로문제(Vehicle Routing Problem, VRP)와는 달리 수집을 위한 고객(backhaul customer)과 배달을 위한 고객(linehaul customer)이 서로 나누어져 있

어 한 경로에서 수집과 배달을 동시에 수행한다. 그러나 한 고객을 방문해 수집과 배달을 동시에 할 경우 작업이 복잡해지므로 수집하기 전에 모든 배달이 이루어져야 한다고 가정한다. 따라서 가능한 운송경로가 되기 위해서는 배달을 위한 고객을 우선 방문한 후 수집을 위한 고객을 방문해



야 하며, 배달물량이나 수집물량이 차량의 용량을 넘어서는 안 된다. 차량대수나 총 운송거리 또는 총 운송비용의 최소화가 목적함수가 된다. (그림 3)은 backhaul이 있는 차량경로문제의 예를 보여준다.

고객을 방문하는데 있어 시간제한이 있을 경우 VRPB는 VRPBTW가 된다. VRPBTW는 각 고객마다 서비스를 허용하는 작업시간대가 정해져 있어서 각 차량은 이 시간 내에 서비스를 해야 한다. 운송계획에서 고려하는 교환센터 경유경로의 경우 교환센터와 각 집중국에 시간제한이 존재하며 출발 집중국에서 물량을 신고 교환센터로 간 후 다시 원래 출발했던 집중국으로 물량을 가져오기 때문에 배달과 수집이 한 경로에서 이루어진다고 할 수 있다. 따라서 VRPBTW의 한 유형이라고 볼 수 있다. VRPBTW의 경우 수집을 위한 고객과 배달을 위한 고객이 따로 존재하며 하나의 경로는 센터에서 출발해 센터로 돌아온다. 이와 달리 교환센터 경유경로는 집중국에서 출발해 교환센터를 방문한 후 다시 집중국으로 돌아오며 출발지와 도착지가 서로 같은 집중국이라는 점에서 차이점이 존재한다. 배달이나 수집 중 한 가지만 할 경우 이 문제는 시간제한이 있는 차량경로 문제((Vehicle Routing Problem with Time Windows, VRPTW)가 되며, VRPTW가 NP-hard [Savelsbergh, 1985]이므로 VRPBTW도 NP-hard이다.

VRP[Gendreau et al. 1994, Osman 1993], VRPB[Goetschackx & Jacobs-Blecha 1989, Mingozzi et al. 1999, Toth & Vigo 1997, Toth & Vigo 1999]에 대해 예전부터 많은 연구가 이루어지고 있는 반면, VRPBTW에 대한 연구는 주로 최근에 이루어지고 있다. 휴리스틱 해법, 최적화 기법, 타부 탐색 등과 같은 메타 휴

리스틱을 이용한 기법들이 제시되고 있다. Thangiah [1994]는 경로 발생 휴리스틱과 경로 개선 휴리스틱의 2단계 기법(two-phase method)을 적용해 국소탐색(local search) 휴리스틱을 제안했다. Gelinas [1995]와 Yano [1987]는 분지한계법(branch and bound)을 이용한 최적화 기법을 제시했으며, Duhamel [1997]은 타부 탐색을 이용한 해법을 제시한 바 있다. Duhamel[1997]은 초기 가능 해를 발생시키기 위해 Solomon [1987]의 insertion 휴리스틱을 변형한 greedy insertion 휴리스틱을 이용하고 여기서 생성된 해에 타부 탐색 휴리스틱을 적용하였다.

VRP에 대해 타부 탐색과 simulated annealing을 적용한 결과 타부 탐색을 적용했을 때 더 좋은 해를 제공하였다 [Osman 1993]. 또한 VRPBTW에 대해 유전 알고리즘(genetic algorithm)과 타부 탐색을 적용한 결과 타부 탐색이 더 좋은 해를 제공하였으며, 유전 알고리즘의 경우 타부 탐색과 비교해 해를 구하는데 시간이 훨씬 오래 걸렸다 [Duhamel 1997]. 본 연구에서는 이러한 결과들에 근거하여 사전처리 작업 후의 운송계획 문제에 대해 타부 탐색 기법을 적용하고자 한다.

### 3. 수리 모형

3장에서는 본 논문에서 다루는 운송계획 문제에 대한 이해를 돕기 위해 정식화를 한다. 2장에서 언급한 사전처리 작업을 한 후의 운송계획 문제에 대한 notation과 수리 모형 (formulation)은 다음과 같다.

[Notation]

$V$ : 운송국 집합,  $V = V_0 \cup V_1 \cup V_2$

$V_0$ : 교환센터 집합,  $V_0 = \{0\}$

$V_1$ : 상행경로상의 집중국 집합,

$$V_1 = \{1, 2, \dots, N\}$$

$V_1'$ : 상행경로상의 집중국 중에서 발생물  
량이 있는 집중국 집합,  $V_1' \subseteq V_1$

$V_2$ : 하행경로상의 집중국 집합,

$$V_2 = \{1', 2', \dots, N'\}$$

$V_2'$ : 하행경로상의 집중국 중에서 도착물  
량이 있는 집중국 집합,  $V_2' \subseteq V_2$

$E$ : 운송로 집합

$Q$ : 트럭유형 집합

$P_i$ : 집중국  $i \in V_1$ 에서 출발해 교환센터로  
가는 가능한 상행경로 집합

$P$ : 가능한 상행경로 집합,  $P = \bigcup_{i \in V_1} P_i$

$P_i'$ : 교환센터에서 출발해 집중국  $i' \in V_2$   
에 도착하는 가능한 하행경로 집합

$P'$ : 가능한 하행경로 집합,

$$P' = \bigcup_{i' \in V_2} P_i'$$

$c_p^q$ : 경로  $p \in P \cup P'$ 를 트럭유형  $q \in Q$ 로 운  
행할 때 운송비용

$$a_{ip} = \begin{cases} 1 & \text{집중국 } i \in V_1' \cup V_2' \text{의 물량이} \\ & \text{경로 } p \in P \cup P' \text{에 의해 운송되는} \\ & \text{경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

[Decision variables]

$$x_p^q = \begin{cases} 1 & \text{트럭유형이 } q \in Q \text{인 상행경로} \\ & p \in P \text{를 선택하는 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

$$y_{p'}^q = \begin{cases} 1 & \text{트럭유형이 } q \in Q \text{인 하행경로} \\ & p' \in P' \text{를 선택하는 경우} \\ 0 & \text{그렇지 않은 경우} \end{cases}$$

[Formulation]

$$\min \sum_{p \in P} \sum_{q \in Q} c_p^q x_p^q + \sum_{p' \in P'} \sum_{q \in Q} c_{p'}^q y_{p'}^q$$

$$\text{s.t. } \sum_{p \in P} \sum_{q \in Q} a_{ip} x_p^q \geq 1, \forall i \in V_1' \quad (10)$$

$$\sum_{p \in P'} \sum_{q \in Q} a_{i'p'} y_{p'}^q \geq 1, \forall i' \in V_2' \quad (11)$$

$$\sum_{p \in P_i} x_p^q - \sum_{p' \in P_i'} y_{p'}^q = 0, \\ \forall i \in V_1, \forall i' \in V_2, \forall q \in Q \quad (12)$$

$$x_p^q \in \{0, 1\}, \forall p \in P, \forall q \in Q$$

$$y_{p'}^q \in \{0, 1\}, \forall p' \in P', \forall q \in Q$$

위 수리 모형에서 목적식은 총 운송비  
용을 최소화하는 것을 의미하며, 총 운송비  
용은 상행경로의 운송비용 합과 하행경로  
의 운송비용 합을 더한 값이 된다. 제약식  
(10)은 상행경로 상의 모든 집중국을 한 번  
이상씩 방문해야 한다는 것을 의미하고, 제  
약식 (11)은 하행경로 상의 모든 집중국을  
한 번 이상씩 방문해야 한다는 것을 의미  
한다. 집중국  $i \in V_1 \cup V_2$ 의 물량이 없는 경  
우에는 경로  $p \in P \cup P'$ 가 그 집중국을 경  
유하더라도  $a_{ip} = 0$ 이므로 제약식 (10)과 제  
약식 (11)에서는 물량이 있는 집중국만을  
고려한다. 제약식 (12)는 출발 집중국과 도  
착 집중국이 같고 상행경로와 하행경로의  
트럭유형이 일치해야 한다는 것을 의미한  
다.

위 수리 모형의 경우 변수가 되는 가능  
한 경로들의 수는 지수적(exponentially)으  
로 많아서 처음부터 모든 변수를 포함시켜  
문제를 풀 수 없다. 그러나 위 모형과 같이  
변수의 수가 지수적으로 많은 경우 선형계  
획 완화문제(LP relaxation)을 열생성  
(column generation) 기법을 사용해 해결할  
수 있다. 본 운송계획 문제의 경우 가능한  
경로들을 생성하는 열생성 문제는 시간 제  
한과 용량 제한이 있는 최단경로문제로서  
NP-hard임이 알려져 있다 [Ball et al.  
1995]. 뿐만 아니라, 각 경로의 운송비용은  
비선형(nonlinear)인 운송비용 테이블로부  
터 구해지므로 열생성 문제의 목적식은 비  
선형이 된다. 따라서 본 연구에서는 위 모

형을 이용하여 최적 해를 구하는 알고리즘 대신 휴리스틱 알고리즘을 개발하고자 한다.

## 4. 타부 탐색 알고리즘

사전처리 작업 후의 운송계획 문제를 해결하기 위한 알고리즘은 insertion 휴리스틱을 이용해 가능 해를 생성하는 해의 발생 단계와 해의 발생 단계에 의해 생성된 가능 해를 타부 탐색을 이용해 개선하는 해의 개선 단계로 이루어진다.

### 4.1 해의 발생 단계

해의 발생 단계는 타부 탐색에 적용할 초기 가능 해를 생성하는 단계로 본 연구에서는 Solomon [1987]의 insertion 휴리스틱을 변형하여 적용한다. Solomon의 insertion 휴리스틱은 경로를 하나씩 생성해 가는 방법으로 하나의 노드를 선택해 센터에서 노드로, 노드에서 다시 센터로 이어지는 초기 경로를 생성한 뒤 이 경로에 다른 노드를 하나씩 insert하는 방법이다. Insert할 노드를 선택하는 기준은 다양하며 이익을 최대화하거나 비용의 감소를 최대화 하는 것 등을 예로 들 수 있다.

본 운송계획에서 고려하는 교환센터 경유경로의 경우 출발 집중국과 도착 집중국이 같아야 하기 때문에 각 집중국을 하나씩 insert하는 방법은 적용이 불가능하다. 또한 하나의 집중국이 insert되었을 때 비용의 감소나 이익을 계산하는 방법에도 정확한 기준이 없다. 따라서 출발 집중국과 도착 집중국이 같아야 한다는 조건을 만족시키기 위해 집중국-교환센터-집중국 형태의 초기 경로를 생성하고, 집중국을 하나씩 insert하는 방법 대신 경로 단위로 insert를

수행한다. 즉 한 경로상의 모든 집중국을 다른 경로에 insert하여 경로 수를 줄이고자 하였다. 또한 insert를 수행한 후 수행전의 총 운송비용과 비교함으로써 어떤 해가 더 나은지를 판단할 수 있게 하였다. 해의 발생 단계의 흐름도는 (그림 4)와 같으며, 알고리즘 단계는 다음과 같다.

[해의 발생 단계]

단계 1 집중국-교환센터-집중국 형태의 초기 가능 해  $S$ 를 생성한다.

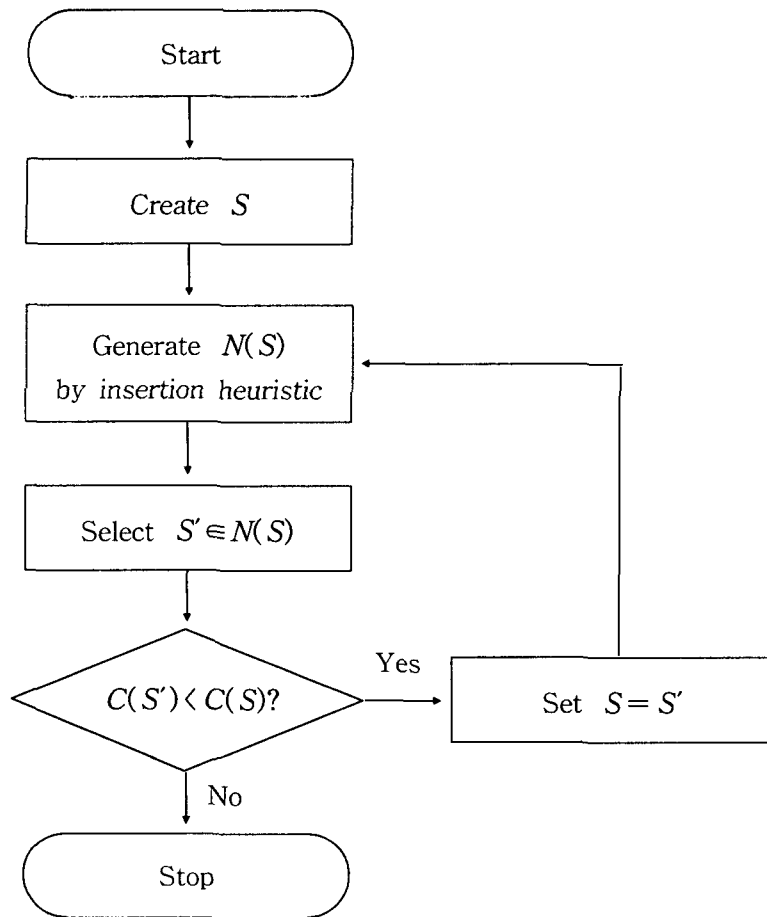
단계 2  $S$ 에 속하는 각 경로를 다른 모든 경로에 insert하여 이웃 해 집합  $N(S)$ 를 생성한다.

단계 3 새로운 가능 경로해  $N(S)$  중 총 운송비용이 최소인 가능해  $S'$ 을 선택하여  $C(S') < C(S)$ 인 경우 단계 4로 가며, 그렇지 않은 경우 종료한다.

단계 4  $S=S'$ 으로 한 후 단계 2로 간다.

단계 1에서는 사전처리 작업에 의해 물량이 분할된 상태에서 각 집중국별로 출발 집중국-교환센터-도착 집중국으로 연결되는 초기 교환센터 경유경로를 생성한다. 초기 경로의 경우 2장에서 언급한 시간 제한 가정과 비분할 선적 가정에 의해 제약조건을 만족하기 때문에 가능 해가 된다. 예를 들어 상행경로상의 집중국들의 집합을  $V_1 = \{1, 2, \dots, N\}$ , 하행경로상의 집중국들의 집합을  $V_2 = \{1', 2', \dots, N'\}$ 라 할 경우, 초기 가능 해  $S = \{R_1, \dots, R_M\}$ 의 각 경로  $R_i (i \in \{1, 2, \dots, M\})$ 가 방문하는 운송국들의 집합은  $R_i = \{i, 0, i'\}$ 과 같다.

단계 2에서는 가능 해  $S$ 에 속하는 각 경로에 대해 insert를 수행하여 이웃 해 집합  $N(S)$ 를 생성한다. Insert란 한 경로상의



(그림 4). 해의 발생 단계의 흐름도.

모든 운송국들을 다른 경로에 포함시키는 것으로 insert에 의해 경로 수가 감소하게 된다. 이 때 경로  $R_i$ 의 물량에 따라 세 가지 경우가 존재한다.  $R_i$ 의 상행물량이 없는 경우 하행경로상의 모든 집중국을 다른 모든 하행경로의 가능한 위치에 insert하는 것을 고려한다. 출발 집중국과 도착 집중국이 같아야 하기 때문에 가능한 위치란 교환센터와 도착 집중국 사이를 의미한다. 경로  $R_i$ 의 하행물량이 없는 경우 상행경로상의 모든 집중국을 다른 모든 상행경로의 가능한 위치에 insert하는 것을 고려하며, 가능한 위치란 출발 집중국과 교환센터 사이를 의미한다. 경로  $R_i$ 의 상하행물량이 모두 있는 경우 상행경로와 하행경로를 모

두 insert하는 것을 고려한다. 이 경우 상행 경로와 하행경로는 서로 같은 경로에 insert될 수도 있고 서로 다른 경로에 insert될 수도 있다. insert된 후의 새로운 경로가 시간제한이나 차량용량 등의 제약 조건을 만족시키지 못하는 경우 고려 대상에서 제외한다. 예를 들어 가능 해  $S = \{R_1, \dots, R_N\}$ 중에서 상하행물량이 모두 있는 경로  $R_i = \{i_1(1), \dots, i_1(m_i), 0, i_2(1), \dots, i_2(n_i)\}$ 를 insert하는 것을 고려할 경우  $R_i$ 를 제외한  $S$ 의 모든 경로에 대해 식 (13)과 같이 순서를 정한다.

$$\begin{aligned} & (R_1, R_1), (R_1, R_2), \dots, (R_1, R_N), (R_2, R_1), \\ & (R_2, R_2), \dots, (R_2, R_N), \dots, (R_N, R_N) \end{aligned} \quad (13)$$

식 (13)에서 (A,B)의 A는  $R_i$ 의 상행경로를 insert할 경로를 의미하며, B는  $R_i$ 의 하행경로를 insert할 경로를 의미한다.  $R_i$ 의 상행경로와 하행경로를 모두  $R_j$ 에 insert할 경우 상행경로의 가능한 위치는 출발 집중국  $j_1(1)$ 과 교환센터 사이에 있는 경유지들의 사이가 되고, 하행경로의 경우 교환센터와 도착 집중국  $j_2(n)$  사이에 있는 경유지들의 사이가 된다. 이웃 해 집합  $N(S)$ 란 insert에 의해 생성된 모든 경로 해  $S'$ 의 집합을 의미한다.

단계 3에서는 단계 2에서 생성된 이웃 해 집합  $N(S)$  중 총 운송비용이 최소인 가능 해  $S'$ 을 선택한다.  $S'$ 의 총 운송비용이 현재 해  $S$ 의 총 운송비용보다 작은 경우 ( $C(S') < C(S)$ ) 단계 4로 가며, 그렇지 않은 경우 총 운송비용을 더 이상 감소시키지 못하므로 해의 발생 알고리즘을 종료한다.

단계 4에서는 현재 해  $S$ 를 가능 해  $S'$ 으로 놓고 ( $S=S'$ ), 단계 2로 가서 위의 과정을 반복한다.

## 4.2 타부 탐색

해의 개선 단계에서는 해의 발생 단계에서 insertion에 의해 생성된 가능 해를 타부 탐색을 이용해 개선한다. 타부 탐색은 국소 해에 머물지 않고 종료 조건을 만족할 때까지 해를 찾는 과정을 계속하기 때문에 적어도 국소 해와 같거나 더 나은 해를 찾게 된다.

타부 탐색에서는 매 반복횟수(iteration)

마다 이웃 해 집합  $N_i(S)$ 를 생성해 이 중에서 목적함수가 가장 많이 개선되거나, 개선되는 해가 없는 경우 목적함수의 증가가 가장 적은 해를 선택한다. 개선되지 않는 해를 선택하는 경우 이전의 해로 다시 돌아가 순환(cycle)이 발생할 수 있으므로 이를 방지하기 위해 타부 제한을 둔다. 즉 새로운 가능 해를 생성한 후 이 해가 타부가 아닌 경우에 한해  $N_i(S)$ 에 포함하며, 타부인지 아닌지를 확인하기 위해서 타부 목록 ( $TABL$ , tabu list)을 이용한다. 본 타부 탐색에서는 타부 목록에 어떤 집중국이 어떤 경로로부터 몇 번째 반복횟수에서 빠져나왔는지를 저장하여 원래 해 상태로 돌아가는 것을 방지한다.

타부 목록에 금지된 이동을 저장한 후 연속  $|T_s|$ 번의 반복횟수(tabu list size)이 후에 타부 목록에서 제외해야 한다 [Osman 1993]. 타부를 확인하는 방법으로 집중국  $i$ 가 경로  $R_a$ 에서 제외되고 집중국  $j$ 가 경로  $R_b$ 에서 제외된 경우 이후  $k$ 번째 반복횟수일 때 식 (14) 중 하나를 만족하면 타부가 된다.  $TABL(i, R_a)$ 는 집중국  $i$ 가 경로  $R_a$ 에서 제외되었을 때의 반복횟수를 의미하고,  $TABL(j, R_b)$ 는 집중국  $j$ 가 경로  $R_b$ 에서 제외되었을 때의 반복횟수를 의미한다.

$$\begin{aligned} k - TABL(i, R_a) < |T_s|, \\ k - TABL(j, R_b) < |T_s| \end{aligned} \quad (14)$$

순환을 방지하기 위한 타부 제약은 좋은 해가 있을 수 있는 곳으로 움직이는 것을 방해할 수 있다. 이러한 한계를 극복하기 위한 새로운 조건이 열망기준(aspiration criteria)이다. 즉 새로운 가능 경로해  $S'$ 이 타부라 하더라도 지금까지 발견된 가장 좋

은 해  $S_b$ 보다 더 좋은 경우  $(C(S') < C(S_b))$  타부 상태를 무시하고 새로운 해로 받아들여 순환을 방지하면서 더 좋은 해를 찾는 것을 가능하게 한다.

본 연구에서는 이웃 해 집합  $N_\lambda(S)$ 를 생성하는 방법으로  $\lambda$ -interchange[Toth & Vigo 1999]를 변형하여 사용한다. 가능해  $S = \{R_1, \dots, R_p, \dots, R_q, \dots, R_N\}$ 가 주어졌을 때  $R_p$ 는 경로  $p$ 가 방문하는 운송국들의 집합이다. 운송국들의 집합  $R_p$ 와  $R_q$  사이에  $\lambda$ -interchange를 적용할 경우, 크기가 각각  $\lambda$ 보다 작거나 같은  $R_p$ 의 부분집합  $S_1$ 과  $R_q$ 의 부분집합  $S_2$ 를 서로 교환한다. 그 결과 경로  $R'_p = (R_p - S_1) \cup S_2$ 과  $R'_q = (R_q - S_2) \cup S_1$ 을 생성해 새로운 가능해  $S' = \{R_1, \dots, R'_p, \dots, R'_q, \dots, R_N\}$ 을 생성한다. 이웃 해 집합  $N_\lambda(S)$ 란  $\lambda$ -interchange에 의해 생성된 모든 가능해  $S'$ 의 집합을 의미한다.  $\lambda$ -interchange를 적용하기 위해서 주어진 가능해  $S = \{R_1, \dots, R_p, \dots, R_q, \dots, R_N\}$ 의 모든 쌍에 대해 식 (15)와 같이 순서를 정한다.

$$\begin{aligned} & (R_1, R_2), (R_1, R_3), \dots, (R_1, R_N), (R_2, R_3), \\ & \dots, (R_2, R_N), \dots, (R_{N-1}, R_N) \end{aligned} \quad (15)$$

각각의 경로 쌍  $(R_p, R_q)$ 에 대해 운송국들의 집합을 서로 교환한다.  $\lambda = 1$ 인 경우, 1-interchange는 (0, 1), (1, 0), (1, 1)의 operator를 가진다. (0, 1) operator는 경로  $R_q$ 에 속하는 하나의 운송국을  $R_p$ 로 이동하는 것을 의미하며, (1, 0) operator는 반대로 경로  $R_p$ 에 속하는 하나의 운송국을  $R_q$ 로 이동하는 것을 의미한다. (1, 1)의

operator는 경로  $R_p$ 에 속하는 하나의 운송국과 경로  $R_q$ 에 속하는 하나의 운송국을 서로 교환하는 것을 의미한다.

$\lambda$ -interchange에 의해 생성된 이웃 해 집합  $N_\lambda(S)$ 에서 새로운 가능해  $S'$ 을 선택하는 방법에는 BA(Best-admissible)와 FBA(First-best-admissible)의 두 가지가 있다. BA는 목적함수가 가장 많이 개선되거나 개선되는 해가 없는 경우 목적함수의 증가가 가장 적은 해를 선택하는 방법이다. FBA는 greedy approach와 BA를 접목한 방법으로 목적함수가 개선되는 해가 처음으로 생성되는 경우 그 해를 선택하고 개선되는 해가 없는 경우 목적함수의 증가가 가장 적은 해를 선택하는 방법이다.

현재 반복횟수가  $k$ , 해가 개선되지 않은 상태에서의 반복횟수가  $k_b$ 인 경우 타부 탐색의 종료조건은 식 (16)과 같다.

$$k > MAXK, k_b > MAXK_b \quad (16)$$

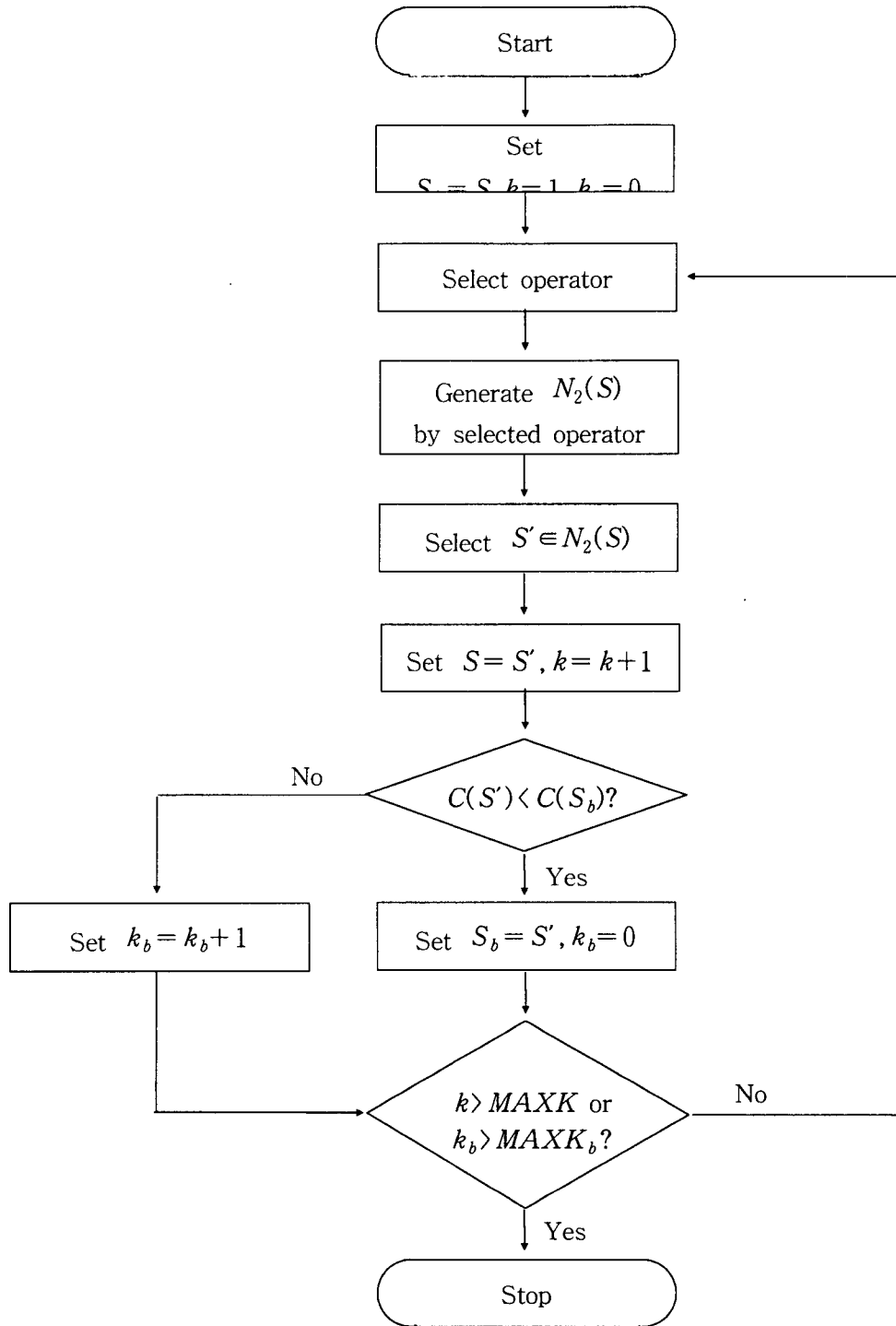
$MAXK$ 는 최대 반복횟수를 의미하고,  $MAXK_b$ 는 해가 개선되지 않은 상태에서의 최대 반복횟수를 의미한다. 즉  $k$ 가  $MAXK$ 보다 크거나  $k_b$ 가  $MAXK_b$ 보다 큰 경우 타부 탐색을 종료한다.

타부 탐색의 흐름도는 (그림 5)와 같으며, 알고리즘 단계는 다음과 같다.

[타부 탐색]

단계 1  $S_b = S, k = 1, k_b = 0$ 으로 한다.

단계 2  $S$ 에 속하는 각 경로에 대해 (1, 0), (1, 1), (2, 0) 중 random하게 선택한 operator를 적용하여 이웃 해 집합  $N_2(S)$ 를 생성한다. 선택 방법에 따라  $N_2(S)$  중 가능해



(그림 5). 타부 탐색의 흐름도.

$S'$ 을 선택해  $S = S'$ 으로 하고  
 $k = k + 1$ 로 한다.  $C(S') < C(S_b)$ 인  
 경우  $S_b = S', k_b = 0$ 으로 하며, 그

렇지 않은 경우  $k_b = k_b + 1$ 로 한  
 다.  
 단계 3  $k > MAXK$  또는  $k_b > MAXK_b$ 인 경

우 단계 4로 가며, 그렇지 않은 경우 단계 2로 간다.

단계 4 알고리즘을 종료하고,  $S_b$ 를 최종 해로 결정한다.

단계 1에서는 해의 발생 단계에 의해 생성된 해  $S$ 를 현재까지 가장 좋은 해  $S_b$ 라 한다. 반복횟수를 나타내는 변수  $k$ 를 1로 하고, 해가 개선되지 않은 상태에서의 반복횟수를 나타내는 변수  $k_b$ 를 0으로 한다. 반복(iteration)이란 단계 2에 의해 이웃 해 집합  $N_2(S)$ 를 생성하고 그 중 가능 해  $S'$ 을 선택해  $S$ 를  $S'$ 으로 놓는 과정을 말한다.

단계 2에서는 현재 해  $S$ 로부터 이웃 해 집합  $N_2(S)$ 를 생성한다.  $N_2(S)$ 를 생성하기 위해 본 알고리즘에서는  $\lambda$ -interchange를 변형하여 적용한다.  $\lambda=2$ 인 경우, 2-interchange는 (0, 1), (1, 0), (1, 1), (0, 2), (2, 0), (1, 2), (2, 1), (2, 2)의 operator를 가진다. 단계 2에서는 이 중에서 (1, 0), (1, 1), (2, 0)의 operator만을 고려하여 이 중에서 random하게 선택해 이웃 해 집합  $N_2(S)$ 를 생성한다. (1, 0) operator는 (0, 1) operator를 포함하고, (2, 0) operator는 (0, 2) operator를 포함하며 (1, 2), (2, 1), (2, 2) operator의 경우 해를 개선시킬 가능성이 적다고 판단하여 고려 대상에서 제외하였다. BA와 FBA의 2가지 방법 중 선택한 방법에 의해  $N_2(S)$  중 가능 해  $S'$ 을 선택한다. 현재 가능 해  $S$ 를  $S'$ 으로 하고, 반복횟수를 하나 증가시킨다. 새로 선택한 가능 해  $S'$ 의 총 운송비용이  $S_b$ 의 총 운송비용보다 작은 경우  $S_b=S'$ 으로 하고 해가 개선되었으므로  $k_b=0$ 으로

한다. 그렇지 않은 경우  $k_b=k_b+1$ 로 한다.

단계 3에서는 알고리즘의 종료 조건을 확인한다.  $k$ 가  $MAXK$ 보다 크거나  $k_b$ 가  $MAXK_b$ 보다 큰 경우 단계 4로 가며, 그렇지 않은 경우 단계 2로 가서 위의 과정을 반복한다.

단계 4에서는 종료 조건을 만족하기 때문에 알고리즘을 종료하고, 현재 가장 좋은 해  $S_b$ 를 최종 해로 결정한다.

## 5. 전산 실험 결과

### 5.1 실험 문제

실제 문제의 경우 교환센터와 집중국들을 포함한 운송국 수가 22개이고 시간 제한이 4시간인 점을 고려해 예제 문제로 운송국 수가 11개, 16개, 21개이고 시간 제한이 3시간, 4시간인 경우에 대해 각각 5문제씩 총 30문제를 생성하였다. 시간 제한이란 상행경로에서 출발 집중국의 출발가능시각과 교환센터의 도착한계시각과의 차이와 하행경로에서 교환센터의 출발가능시각과 도착 집중국의 도착한계시각의 차이를 의미한다.

운송로 그래프는 양방향 완전그래프(complete graph)이며 모든 운송로의 거리는 유클리디언 거리(Euclidean distance)이고, 철도 운송은 없다고 가정하였다. 실제 전국 운송계획 문제의 데이터와 같이 트럭 유형으로는 2.5t, 4.5t, 8t의 세 가지를 고려하였고 트럭의 공통속도는 80km/h, 트럭 유형별 적재비율은 모두 70%로 하였다. 트럭 유형별로 트럭대수의 상한 및 트럭의 고정비용은 없으며, 모든 운송경로에서 트럭의 탑승인원은 1인으로 가정하였다. 자세한 트럭 적재유형 테이블과 운송비용 테이블 자



료 등은 [최지영 2001]에 제시된 자료를 사용하였다.

운송물량은 통상과 소포로 나누어 생성하였다. 통상과 소포 모두 물량단위는 통으로 하고 중간단위는 행낭으로, 최종단위는 팔레트로 환산하였다. 실제 문제의 데이터를 고려하여 통상물량은 구간  $U[50000, 150000]$ 에서 생성하고 소포물량은 구간  $U[100, 300]$ 에서 생성하였다. 타부 탐색 알고리즘에서는 알고리즘의 특성을 고려하여 필요한 파라미터 값으로  $|T_s|=5, MAXK=5000, MAXK_b=3000$ 을 사용하였다.

## 5.2 실험 결과

본 연구에 의해 제시된 알고리즘은 java로 구현되었으며, Pentium (700MHz) PC에서 실행되었다. 타부 탐색을 하는 과정에 random 요소가 있기 때문에 BA와 FBA를 적용하여 각각 10번씩 실행해 그 결과의 평균을 취하였다. 운송국 수가 11개

인 문제의 경우 가능한 상행경로와 하행경로를 모두 생성하여 3장에서 제시한 수리모형을 이용해 CPLEX 6.0에 의해 최적해를 구했으며, 운송국 수가 16개인 경우나 21개인 문제의 경우 가능한 경로의 수가 너무 많아 최적해를 구하는 것이 불가능하였다.

운송국 수가 11개인 문제에 한해서 모든 가능한 경로의 집합을 이용해 CPLEX에 의해 얻은 최적해와 비교를 하였다. 그 결과 BA를 적용한 타부 탐색과 FBA를 적용한 타부 탐색의 경우 'GAP'이 각각 0.287%와 0.508%로 타부 탐색 결과 얻어진 해의 품질이 매우 좋음을 알 수 있었으며, BA를 적용한 경우가 FBA를 적용했을 때보다 약간 더 좋은 결과를 주었다. 운송국 수가 16개, 21개인 문제의 경우 가능한 경로의 수가 너무 많아 최적해를 구하는 것이 불가능하여 구해진 해의 품질을 비교할 수 없었으나, 운송국 수가 11개인 문제의 경우와 같이 그 결과가 나쁘지 않을 것으로 생각된다.

<표 1>. 문제의 특성

| Problem number | # of nodes | Time limit(h) | Problem number | # of nodes | Time limit(h) | Problem number | # of nodes | Time limit(h) |
|----------------|------------|---------------|----------------|------------|---------------|----------------|------------|---------------|
| 113_1          | 11         | 3             | 163_1          | 16         | 3             | 213_1          | 21         | 3             |
| 113_2          | 11         | 3             | 163_2          | 16         | 3             | 213_2          | 21         | 3             |
| 113_3          | 11         | 3             | 163_3          | 16         | 3             | 213_3          | 21         | 3             |
| 113_4          | 11         | 3             | 163_4          | 16         | 3             | 213_4          | 21         | 3             |
| 113_5          | 11         | 3             | 163_5          | 16         | 3             | 213_5          | 21         | 3             |
| 114_1          | 11         | 4             | 164_1          | 16         | 4             | 214_1          | 21         | 4             |
| 114_2          | 11         | 4             | 164_2          | 16         | 4             | 214_2          | 21         | 4             |
| 114_3          | 11         | 4             | 164_3          | 16         | 4             | 214_3          | 21         | 4             |
| 114_4          | 11         | 4             | 164_4          | 16         | 4             | 214_4          | 21         | 4             |
| 114_5          | 11         | 4             | 164_5          | 16         | 4             | 214_5          | 21         | 4             |

<표 2>. 타부 탐색 휴리스틱 결과 (BA,  $n=11$ )

| Problem number | Optimal value | Total iter. avg. | Best iter. avg. | CPU time (sec) | Solution value | GAP (%) |
|----------------|---------------|------------------|-----------------|----------------|----------------|---------|
| 113_1          | 1784404       | 3848.9           | 850.6           | 94.9           | 1784404.0      | 0.00    |
| 113_2          | 1347221       | 4347.6           | 1465.0          | 66.0           | 1354491.0      | 0.54    |
| 113_3          | 1757262       | 3453.6           | 451.6           | 23.8           | 1765749.2      | 0.48    |
| 113_4          | 2346441       | 4259.5           | 1813.9          | 91.3           | 2365302.8      | 0.80    |
| 113_5          | 1098928       | 3434.9           | 432.9           | 27.9           | 1098928.0      | 0.00    |
| 114_1          | 1815850       | 3865.0           | 1150.0          | 44.4           | 1820364.8      | 0.25    |
| 114_2          | 1617520       | 4441.2           | 1748.3          | 89.9           | 1622101.1      | 0.28    |
| 114_3          | 1837778       | 4250.4           | 1610.9          | 101.2          | 1841460.8      | 0.20    |
| 114_4          | 1861794       | 4228.2           | 1604.8          | 72.9           | 1862961.2      | 0.06    |
| 114_5          | 1333616       | 3351.9           | 349.9           | 35.1           | 1337060.9      | 0.26    |
| Avg.           |               |                  |                 |                |                | 0.287   |

<표 3>. 타부 탐색 휴리스틱 결과 (BA,  $n=16$ )

| Problem number | Optimal value | Total iter. avg. | Best iter. avg. | CPU time (sec) | Solution value | GAP (%) |
|----------------|---------------|------------------|-----------------|----------------|----------------|---------|
| 163_1          | *             | 3036.4           | 34.4            | 137.9          | 2713769.6      | *       |
| 163_2          | *             | 4296.9           | 1654.9          | 154.9          | 2285829.3      | *       |
| 163_3          | *             | 4051.9           | 1500.8          | 144.7          | 1999584.0      | *       |
| 163_4          | *             | 4138.8           | 1250.0          | 161.1          | 1746570.4      | *       |
| 163_5          | *             | 4435.7           | 1769.9          | 183.6          | 1713108.6      | *       |
| 164_1          | *             | 4214.1           | 1400.4          | 131.6          | 1792359.4      | *       |
| 164_2          | *             | 4022.7           | 1429.3          | 124.0          | 1830960.4      | *       |
| 164_3          | *             | 3378.2           | 386.7           | 118.9          | 1351899.6      | *       |
| 164_4          | *             | 4826.4           | 2211.5          | 252.6          | 2016230.2      | *       |
| 164_5          | *             | 3457.5           | 628.1           | 113.4          | 2352425.8      | *       |
| Avg.           |               |                  |                 |                |                | *       |

<표 4>. 타부 탐색 휴리스틱 결과 (BA,  $n=21$ )

| Problem number | Optimal value | Total iter. avg. | Best iter. avg. | CPU time (sec) | Solution value | GAP (%) |
|----------------|---------------|------------------|-----------------|----------------|----------------|---------|
| 213_1          | *             | 4161.4           | 1180.7          | 583.7          | 2948287.7      | *       |
| 213_2          | *             | 4230.0           | 1900.8          | 648.5          | 3374512.7      | *       |
| 213_3          | *             | 4852.6           | 3051.0          | 740.6          | 3064595.1      | *       |
| 213_4          | *             | 4671.0           | 2959.7          | 1061.3         | 4128888.0      | *       |
| 213_5          | *             | 4433.0           | 1984.2          | 677.2          | 4040948.1      | *       |
| 214_1          | *             | 4125.0           | 1399.6          | 956.7          | 3993514.3      | *       |
| 214_2          | *             | 3521.7           | 607.8           | 461.9          | 3091633.6      | *       |
| 214_3          | *             | 3705.7           | 917.2           | 981.1          | 3345790.4      | *       |
| 214_4          | *             | 4748.5           | 2456.0          | 1276.4         | 4410230.8      | *       |
| 214_5          | *             | 4391.2           | 1823.6          | 814.1          | 3931707.5      | *       |
| Avg.           |               |                  |                 |                |                | *       |

<표 5>. 타부 탐색 휴리스틱 결과 (FBA,  $n=11$ )

| Problem number | Optimal value | Total iter. avg. | Best iter. avg. | CPU time (sec) | Solution value | GAP (%) |
|----------------|---------------|------------------|-----------------|----------------|----------------|---------|
| 113_1          | 1784404       | 3673.9           | 679.9           | 64.5           | 1784787.8      | 0.02    |
| 113_2          | 1347221       | 4206.1           | 1737.9          | 44.1           | 1360620.8      | 0.99    |
| 113_3          | 1757262       | 3856.8           | 971.9           | 17.9           | 1765496.4      | 0.47    |
| 113_4          | 2346441       | 3775.8           | 883.0           | 56.8           | 2377720.4      | 1.33    |
| 113_5          | 1098928       | 4476.4           | 1634.0          | 25.1           | 1101607.0      | 0.24    |
| 114_1          | 1815850       | 3638.6           | 1102.7          | 30.1           | 1821772.7      | 0.33    |
| 114_2          | 1617520       | 4705.9           | 3120.4          | 69.8           | 1624713.8      | 0.44    |
| 114_3          | 1837778       | 3974.7           | 1405.3          | 68.4           | 1846550.9      | 0.48    |
| 114_4          | 1861794       | 4096.3           | 1542.9          | 49.4           | 1868326.0      | 0.35    |
| 114_5          | 1333616       | 3575.7           | 573.7           | 26.7           | 1339371.2      | 0.43    |
| Avg.           |               |                  |                 |                |                | 0.508   |

<표 6>. 타부 탐색 휴리스틱 결과 (FBA,  $n=16$ )

| Problem number | Optimal value | Total iter. avg. | Best iter. avg. | CPU time (sec) | Solution value | GAP (%) |
|----------------|---------------|------------------|-----------------|----------------|----------------|---------|
| 163_1          | *             | 3004.0           | 2.1             | 103.6          | 2714389.0      | *       |
| 163_2          | *             | 4329.6           | 1515.6          | 105.0          | 2289685.6      | *       |
| 163_3          | *             | 3857.9           | 951.3           | 100.8          | 2007713.7      | *       |
| 163_4          | *             | 3291.0           | 289.0           | 90.4           | 1756279.9      | *       |
| 163_5          | *             | 4351.7           | 1854.5          | 137.6          | 1715327.2      | *       |
| 164_1          | *             | 4208.7           | 1598.9          | 94.6           | 1799258.2      | *       |
| 164_2          | *             | 3853.6           | 890.5           | 78.8           | 1842052.5      | *       |
| 164_3          | *             | 3320.2           | 337.2           | 82.5           | 1350352.2      | *       |
| 164_4          | *             | 4708.8           | 2683.9          | 172.4          | 2013430.4      | *       |
| 164_5          | *             | 3588.9           | 933.9           | 77.8           | 2361615.9      | *       |
| Avg.           |               |                  |                 |                |                | *       |

<표 7>. 타부 탐색 휴리스틱 결과 (FBA,  $n=21$ )

| Problem number | Optimal value | Total iter. avg. | Best iter. avg. | CPU time (sec) | Solution value | GAP (%) |
|----------------|---------------|------------------|-----------------|----------------|----------------|---------|
| 213_1          | *             | 3655.2           | 915.1           | 398.3          | 2947999.8      | *       |
| 213_2          | *             | 3903.1           | 1270.5          | 491.0          | 3370465.4      | *       |
| 213_3          | *             | 4101.1           | 1334.1          | 451.1          | 3086527.7      | *       |
| 213_4          | *             | 4609.8           | 2934.9          | 915.4          | 4133274.6      | *       |
| 213_5          | *             | 3883.3           | 1290.4          | 473.3          | 4066229.0      | *       |
| 214_1          | *             | 4290.9           | 1708.3          | 865.0          | 4008391.2      | *       |
| 214_2          | *             | 3854.3           | 1069.7          | 396.6          | 3091712.6      | *       |
| 214_3          | *             | 4155.6           | 1332.0          | 956.7          | 3349533.7      | *       |
| 214_4          | *             | 4785.5           | 2984.1          | 1068.6         | 4439257.5      | *       |
| 214_5          | *             | 4564.9           | 2092.1          | 687.0          | 3933103.1      | *       |
| Avg.           |               |                  |                 |                |                | *       |

## 6. 결론

본 논문은 허브와 스포크 형태의 운송망에서 하나의 교환센터와 다수의 집중국을 대상으로 운송국들간에 운송경로를 설정하고 설정된 운송경로에 차량을 할당하는 전국 운송계획 수립을 다루고 있다. 운송계획의 목적은 총 운송비용을 최소화하는 것이며 운송계획 수립시 다양한 제약조건을 만족해야 한다.

본 논문에서는 집중국간 직접 운송이나 집중국과 교환센터간 직접 운송에 의해 물량을 사전처리 한 후 남은 물량을 대상으로 운송계획을 수립하였으며, 이에 대한 해법으로 타부 탐색을 이용한 알고리즘을 제시하였다. 알고리즘은 insertion을 이용해 가능 해를 생성하는 해의 발생 단계와 해의 발생 단계에 의해 생성된 가능 해를 타

부 탐색을 이용해 개선하는 해의 개선 단계로 이루어진다. 운송국 수가 11개, 16개, 21개인 문제에 대해 타부 탐색 과정에 BA와 FBA의 두 가지 방법을 적용하여 실행하였으며, 운송국 수가 11개인 문제에 한해 최적해와 비교를 한 결과 GAP이 각각 0.287%와 0.508%로 BA가 FBA보다 약간 더 좋은 결과를 보였다.

추후 연구 방향으로는 simulated annealing이나 유전 알고리즘 등의 다른 메타 휴리스틱을 적용해 결과를 비교해 보거나, 운송국 수가 많아 최적 해를 구할 수 없는 경우 목적식 값에 대한 bound를 구하는 방법을 고려해 볼 수 있다.

## 참고문헌

- [1] 최지영, 우편 운송계획 문제의 타부 탐색 알고리즘, 공학석사 학위논문, 대전:한국과학기술원, 2001, pp.43-45.
- [2] Ball, M. O., T. L. Magnanti, C. L. Monma, G. L. Nemhauser, Network Routing, Amsterdam, North-holland, 1995, pp.1-130.
- [3] Duhamel, C., J.-Y. Potvin and J.-M. Rousseau, "A Tabu Search Heuristic for the Vehicle Routing Problem with Backhauls and Time Windows", Transportation Science, Vol.31, 1997, pp.49-59.
- [4] Gelinas, S., M. Desrochers, J. Desrosiers and M. M. Solomon, "A New Branching Strategy for Time Constrained Routing Problems with Application to Backhauling", Annals of Operations Research, Vol. 61, 1995, pp.91-109.
- [5] Gendreau, M., A. Hertz and G. Laporte, "A Tabu Search Heuristic for the Vehicle Routing Problem", Management Science, Vol. 40, 1994, pp.1276-1290.
- [6] Goetschalckx, M., and C. Jacobs-Blecha, "The Vehicle Routing Problem with Backhauls", European Journal of Operational Research, Vol. 42, 1989, pp.39-51.
- [7] Mingozzi, A., S. Giorgi and R. Baldacci, "An Exact Method for the Vehicle Routing Problem with Backhauls", Transportation Science, Vol.33, 1999, pp.315-329.
- [8] Osman, I. H., "Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem", Annals of Operations Research, Vol. 41, 1993, pp.421-451.
- [9] Savelsbergh, M. W. P., "Local Search for Routing Problems with Time Windows", Annals of Operations Research, Vol.4, 1985, pp.285-305.
- [10] Solomon, M. M., "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints", Operations Research, Vol.35, 1987, pp.254-265.
- [11] Thangiah, S. R., J. Y. Potvin and T. Sun, "Heuristic

Approaches to Vehicle Routing with Backhauls and Time Windows", *Computers & Operations Research*, Vol.23, 1994, pp.1043-1057.

- [12] Toth, P., and D. Vigo, "An Exact Algorithm for the Vehicle Routing Problem with Backhauls", *Transportation Science*, Vol.31, 1997, pp.372-385.
- [13] Toth, P., and D. Vigo, "A Heuristic Algorithm for the Symmetric and Asymmetric Vehicle Routing Problems with Backhauls", *European Journal of Operations Research*, Vol.113, 1999, pp.528-543.
- [14] Yano, C., T. Chan, L. Richter, T. Cutler, K. Murty and D. McGettigan, "Vehicle Routing at Quality Stores", *Interfaces*, Vol.17, 1987, pp.52-63.

## ■ 저자소개



### 최지영

고려대학교 산업공학과에서 학사(1999)와 한국과학기술원 산업공학과에서 공학석사학위(2001)를 취득한 후, 현재까지 한국전자통신연구원 우정기술연구센터 물류기술연구팀에서 연구원으로 재직 중이다. 관심분야는 물류 및 Supply Chain Management이다.



### 송영호

서울대학교 산업공학과에서 학사(1980)와 한국과학기술원 경영과학과 석사(1982), RPI School of Management and Technology 에서 박사(1992) 학위를 취득하였다. 현재 홍익대학교 상경대학 경영정보학과 교수로 재직 중이다. 관심분야는 금융정보, e-비즈니스, simulation, financial engineering 등이다.



### 강성열

서울대학교 산업공학과에서 학사(1981)와 석사(1983), Georgia Tech 산업시스템공학부에서 박사(1992) 학위를 취득하였다. 현재 홍익대학교 상경대학 경영정보학과 교수로 재직 중이다. 관심분야는 정보통신, 인터넷, e-비즈니스, stochastic modeling 등이다.