대한산업공학회/한국경영과학회 2002 춘계공동학술대회
한국과학기술원(KAIST) 2002년 5월 3일-4일

복수 차량 유형에 대한 차량경로문제의 정수계획 해법

최은정 · 이태한 · 박성수*

Integer Programming Approach to the Heterogeneous Fleet Vehicle Routing Problem

Eunjeong Choi, Taehan Lee and Sungsoo Park*
Department of Industrial Engineering, KAIST
*Tel: 042) 869-3121, Fax: 042) 869-3110, E-mail: sspark@kaist.ac.kr

Abstract

We consider the heterogeneous fleet vehicle routing problem (HVRP), a variant of the classical vehicle routing problem (VRP). The HVRP differs from the classical VRP in that it deals with a heterogeneous fleet of vehicles having various capacities, fixed costs, and variables costs. Therefore the HVRP is to find the fleet composition and a set of routes with minimum total cost.

We give an integer programming formulation of the problem and propose an algorithm to solve it. Although the formulation has exponentially many variables, we can efficiently solve the linear programming relaxation of it by using the column generation technique. To generate profitable columns we solve a shortest path problem with capacity constraints using dynamic programming. After solving the linear programming relaxation, we apply a branch-and-bound procedure. We test the proposed algorithm on a set of benchmark instances. Test results show that the algorithm gives best-known solutions to almost all instances.

1. 서론

현대 사회에 있어서 물류가 차지하는 관심과 비중은 생산의 중요성을 넘어서 기업 경쟁력의 척도가 되고 있는 실정이다. 이러한 환경 하에서 전체 물류비의 1/3 ~ 2/3을 차지하는 수송비의 절감은 소비자와 물류 운송회사, 공급자 등으로 이루어지는 공급 사슬(Supply Chain)의 전체 운영비용의 절감으로 이어진다 [1]. 제품을 운송하는 물류 운송회사의 관점에서는 개별 고객들의 수요를 모두 충족시키면서 전체 운송비용을 최소화하고자 함을 목표로 하는데, 이러한 것을 수리적으로 다룬 문제가 차량경로문제(Vehicle Routing Problem, VRP)이다. 즉, 차량경로문제란 한 개 또는 그 이상의 센터를 중심으로 여러 대의 차량이 센터를 출발하여 지역별로 분산된 고객들을 방문하고 다시 센터로 돌아오는 경로를 전체 운송비용이 최소화되도록 결정하는 문제이다. 차량경로문제는 사용되는 차량의 유형, 이동시간의 변화 여부, 공급지의 수, 수집 및 배달의 분리 여부 등에 따른 다양한 형태의 문제들이 정의되고 연구되어 왔다. 본 연구에서는 차량경로문제의 여러 가지 변형 중 복수 차량 유형을 사용하는 HVRP(Heterogeneous Fleet Vehicle Routing Problem)에 대해서 다루고자 한다.

HVRP는 각기 다른 용량을 가진 차량들은 각기 다른 운영비용(고정비, 변동비)이 발생한다고 간주하고, 총 운영비용이 가장 최소가 되는 최적의 차량 조합을 구하는 문제이다. 단, 본 연구에서는 각 차량 유형에 대한 대수의 제한은 없는 것으로 가정한다.

HVRP는 NP-Hard로 잘 알려진 차량경로문제의 일반화된 문제로 역시 NP-Hard이다. 이러한 이유로 대부분의 연구가 최적화 기법을 이용하여 해를 구하는 방법 대신에 발견적 기법(Heuristics)을 적용하여 근사해를 찾는 연구가 활발히 진행되어 왔다. HVRP와 유사한 문제로는 VFSMP(Vehicle Fleet Size and

Mix Problem)가 있으며[9], 이는 VFM (Vehicle Fleet Mix)라고도 불린다[11]. VFM는 HVRP의 특수한 경우로 모든 차량에 상관없이 변동비가 동일하다. VFM을 풀기 위한 발견적 해법으로는 몇 가지 있으며, 이는 대부분 전통적인 차량경로문제의 발견적 해법에서 유도한 것들이다[4][7][8][9][11]. HVRP와 유사한 문제로는 Tailard[12]가 제시한 VRPHE (VRP with a heterogeneous fleet of vehicles)가 있으나, 이는 HVRP와 달리 각 차량 유형에 대한 대수의 제한이 있다. Tailard[12]는 VRPHE를 풀기 위해 타부 서치를 바탕으로 한 열생성(Column Generation) 기법을 제안하였다. 이 방법은 먼저 타부 서치의 AMP(Adaptive Memory Procedure)를 이용하여 각 차량 용량별로 초기해를 구하고 그 해들 중에서 여러 개의 우수한 경로들을 뽑아서 메모리에 저장시킨 뒤 그들의 조합을 이용하여 최소 비용의 차량경로 집합을 구성한다. Gendreau et al.[6]는 본 연구와 동일한 HVRP를 대상으로 하여 타부 서치를 적용한 발견적 해법을 제안하였다. 이 타부 서치 알고리즘은 TSP를 풀기 위해 고안된 GENIUS 삽입 기법을 기반으로 하고 있는데, 랜덤하게 고객들을 경로간에 이동시킨 후 경로 내 지점방문순서를 GENIUS 삽입기법을 이용하여 개선시키는 절차를 반복 수행한다.

본 연구에서는 HVRP에 대한 정수계획(Integer Programming)모형을 제시하고, 이를 풀기 위한 알고리즘을 개발하였다. 우리는 이 모형의 선형계획완화문제(LP Relaxation)를 풀기 위해 열생성 기법을 사용하였다. 열을 생성하기 위해서는 용량 제약이 있는 최단경로문제(Shortest Path Problem)를 동적 프로그래밍(Dynamic Programming)을 이용하여 pseudo-polynomial 시간 안에 풀었다. 선형완화문제를 푼 후에는 하나의 정수해를 찾기 위해 branch-and-bound 기법을 적용하였다.

우리는 Golden et al.[9]에 의해서 제안된 예제 문제들을 대상으로 본 연구에서 제안한 알고리즘을 테스트하였다. 그 결과 대부분의 예제들에 대해서 가장 좋다고 알려진 해를 제공한다는 사실을 알 수 있었다.

본 연구의 구성은 다음과 같다. 2절에서는 HVRP에 대한 정수계획 모형을 제시하고, 3절에서는 알고리즘에 대해서 설명한다. 4절에서는 전산실험을 통하여 제안된 알고리즘의 성능을 분석하고, 마지막으로 5절에서는 결론을 제시한다.

2. 수학적 모형

본 절에서는 HVRP의 정수계획모형을 제시한다.

2.1 Set Partitioning Model

$G=(V,A)$를 하나의 센터와 N개의 고객들의 집합인 V와 두 지점 사이의 호(arc)들의 집합인 A로 이루어진 네트웍이라 하자. 집합 V에서 센터를 0으로, 고객을 $1,\cdots,N$으로 지정하면 각 고객은 수요량 $q_i(i=1,\cdots,N)$를 갖는다. $Q=\{1,\ldots,K\}$를 차량 유형의 집합이라 하면 b_k, f_k, g_k, $k\in Q$는 각각 차량 유형 k의 용량, 고정비, 변동비를 의미한다. 임의의 두 지점 $(i,j)\in A$ 사이의 거리를 d_{ij}라 하자. c_{ijk}는 차량 유형 k가 두 지점 (i,j)를 이동했을 때의 비용을 의미하며, 거리 d_{ij}와 변동비 g_k의 곱으로 구해진다. 그러므로 HVRP는 센터 0에서 출발하여 여러 곳에 흩어져 있는 수요량 q_i를 가진 고객들을 만족시키면서 센터로 돌아오는 최소 비용의 경로들을 구하는 문제이다. 각 고객은 반드시 한번 방문되어야 하며, 한 경로의 총 수요량은 차량 용량 b_k를 초과해서는 안된다.

HVRP는 set partitioning 문제로 모형화 할 수 있다. R을 HVRP에 대한 가능 경로들의 집합이라 하자. 임의의 경로 $r\in R$은 센터를 출발해서 센터로 돌아오며, 그 경로의 총 수요량은 차량 용량 b_k를 초과하지 않는다. δ_{ir}는 경로 r이 고객 i를 방문했을 때는 1, 그렇지 않을 때는 0를 나타내는 상수이다. c_r은 경로 r의 운송비로, 사용된 차량 유형의 고정비와 이동비의 합으로 구해진다. 이 때 이동비는 경로 r를 구성하는 호(arc)들의 비용을 합함으로써 얻을 수 있다.

다음은 위에 설명한 기호들을 가지고 HVRP를 set partitioning 문제로 모형화한 것이다.

$$
\begin{aligned}
(\mathrm{MP}^1)\ \min\ & \sum_{r\in R} c_r x_r \\
s.t\ & \sum_{r\in R} \delta_{ir} x_r = 1, \quad \text{for all } i\in V\setminus\{0\} \qquad (1) \\
& x_r \in \{0,1\}, \qquad \text{for all } r\in R.
\end{aligned}
$$

x_r는 경로 r이 선택되었을 때 1, 그렇지 않을 때는 0을 나타내는 결정 변수이다. 목적함수는 총 운송비를 최소화하는 것이고, 제약식 (1)은 각 고객은 반드시 한번 방문되어야 한다는 것을 의미한다.

MP^1에서 변수로 표현된 열(column)들은 가능 경로들을 의미하며, 그 수는 무수히 많다. 그러므로 모든 가능 경로들을 생성하는 것은 현실적으로 거이 불가능하다. 이러한 이유로 우리는 MP^1의 선형완화문제(MLP^1)를 열생성 기법을 이용해서 푼다. 열생성 기법에서는 초기에 일부의 열을 가지고 제한된 선형완화문제를 풀다가 필요할 때마다 열을 추가하여 풀게 된다. 먼저 우리는 R의 부분 집합 R'이 주어진다고 가정하고, 제한된 MLP^1(RMLP^1)를

구성한다. $RMLP^1$의 i번째 제약식에 해당하는 dual 변수를 π_i라 하자. 우리는 $RMLP^1$를 simplex method를 이용해서 풀 수 있으며, 그 때의 최적해를 z^*, dual 변수값을 π^*라 할 때, 다음 조건을 만족하면 z^*가 MLP^1의 최적해가 된다.

$$\sum_{i\in V\setminus\{0\}} \pi_i^* \delta_{ir} \le c_r, \text{ for all } r\in R\setminus R' \qquad (2)$$

즉, 모든 가능한 경로에 대해서 reduced cost가 0보다 크기 때문에 z^*가 MLP^1의 최적해가 되는 것이다. 그러므로, 열생성 문제는 reduced cost를 최소화하는 가능 경로를 구하는 문제로 다음과 같이 모형화된다.

$$(SP^1)\ \min\ \bar{c}_r = c_r - \sum_{i\in V\setminus\{0\}} \pi_i^* \delta_{ir}$$
$$s.t\ \ r\in R$$

만약 SP^1의 목적값이 음수이면 얻어진 가능 경로는 $RMLP^1$에 열로 추가되며, 변경된 $RMLP^1$는 다시 최적화된다. 그렇지 않으면 최적 조건 (2)를 만족하기 때문에 z^*가 MLP^1의 최적해가 되고, 더 이상 열은 추가되지 않는다.

2.2 열생성 문제에 대한 동적 프로그래밍 모델

열생성 문제인 SP^1은 동적 프로그래밍을 이용해서 풀 수 있다. 이 접근 방법을 소개하기 위해 몇 가지 기호들을 정의한다. $\bar{c}_{ijk}$는 차량 유형 k가 두 지점 (i,j)를 이동했을 때의 reduced cost를 의미하며, 다음과 같이 계산된다.

$$\bar{c}_{ijk} = c_{ijk} - \pi_i, \text{ 단, } \pi_0 = 0.$$

$F_k(S,i)$는 차량 유형 k가 센터에서 집합 S안의 모든 노드들을 단 한번만 방문하면서 노드 i까지 가는 경로의 최소 reduced cost를 의미한다. $a(\sum_{i\in S} q_i)$는 총수요량 $\sum_{i\in S} q_i$를 실을 수 있는 차량 유형들 중 용량이 가장 작은 차량유형을 의미한다. 예를 들어, $k^* = a(\sum_{i\in S} q_i)$이면 $\sum_{i\in S} q_i \le b_{k^*}, k^*\in Q$를 반드시 만족해야 한다. 다음은 SP^1에 대한 동적 프로그램밍 모델을 반복 방정식들(recurrence equations)로 표현한 것이다.

$$F_k(\emptyset, i) = 0, \text{ for all } k\in Q,$$
$$F_{k^*}(S,i) = \min_{(i,j)\in A}\{F_{k^*}(S-\{j\}, i) + \bar{c}_{ijk^*} \mid k^* = a(\sum_{i\in S} q_i)\},$$
$$\text{for all } j\in V\setminus\{0\}, S\subseteq V\setminus\{0\}, \sum_{i\in S} q_i \le b_{k^*}, k^*\in Q.$$

SP^1은 NP-Complete로 알려진 가중치 제약이 있는 최단경로문제(Shortest Weight-constrained Path Problem)를 특수한 경우로 가지고 있다. 그러므로 SP^1은 NP-Hard이다 [5]. 게다가 지금까지 그것을 풀기 위한 psuedo-polynominal 알고리즘도 알려진 것이 없다.

SP^1을 좀 더 쉽게 풀기 위해서는 기본 경로 요건(elementary path requirement)을 완화함으로써 얻을 수 있다. 즉, 경로 내에 사이클을 허용한다는 것을 의미한다. 이렇게 완화된 문제를 SP^2라 하자. SP^2에 대한 동적 프로그래밍 모델은 Christofides, Mingozzi and Toth[2]에 의해 제안되었다. 이 모델을 우리 문제에 적용하기에 앞서 몇 가지 기호들을 정의한다. 먼저 새로운 네트웍 $G' = (V', A')$을 구성한다. V'는 노드 (i,q) $i = 0,1,\ldots,N$, $q = q_i, q_i+1, \ldots, b_K$, 들의 집합이다. A'는 임의의 두 노드 (i,q)와 $(j, q+q_j)$ 사이를 연결하는 호(arc)들의 집합으로, 각 호(arc)들은 reduced cost $\bar{c}_{ijk}$를 가지고 있다. $F_k(i,q)$는 차량 유형 k로 센터에서 노드 (i,q)까지 가는 경로의 최소 reduced cost이다. 이것에 대한 새로운 반복 방정식들은 다음과 같다.

$$F_k(0,0) = 0, \text{ for all } k\in Q,$$
$$F_{k^*}(j,q) = \min_{(i,j)\in A}\{F_{k^*}(i,q') + \bar{c}_{ijk^*} \mid q' + q_j = q, k^* = a(q)\},$$
$$\text{for all } j\in V\setminus\{0\}, q\le b_{k^*}, k^*\in Q.$$

SP^2는 knapsack 문제를 특수한 경우로 가지고 있다. 그러므로 역시 NP-Hard이다. 그러나 그것을 풀기 위한 pseudo-polynomial 알고리즘들이 존재한다.

SP^2는 경로에 사이클을 허용함으로써 SP^1보다 넓은 해공간(solution space)을 갖게 된다. 사이클을 포함하는 경로들을 살펴보면 대부분 형태가 (i,j,i)인 2-사이클을 포함한다. 그러므로 SP^2의 해공간을 줄이기 위해 2-사이클들을 제거하는 것을 고려해 볼 수 있다.

SP^2에서 2-사이클을 제거한 문제를 SP^3라 하자. SP^3에 대한 동적 프로그래밍 모델은 Derochers, Derosiers and Solomon[3]에 의해 제안되었으며, 기본 개념은 네트웍 G'상에서 센터로부터 노드 (i,q)까지의 최적 경로와 그 다음 최적 경로를 각각 유지하는 것이다. $H_k(i,q)$는 차량 유형 k로 센터에서 노드 (i,q)까지 가는 경로의 최소 reduced cost이고, $p(i,q)$는 reduced cost $H_k(i,q)$와 관련된 경로 내에서 노드 (i,q)의 선행자를 의미한다.

$H_k'(i,q)$는 마지막 방문 노드가 $p(i,q)$가 아닌 것들 중에서 센터에서 노드 (i,q)까지 가는 경로의 최소 reduced cost를 의미한다. 명백히 $H_k(i,q) \le H_k'(i,q)$가 성립한다. 다음은 SP[3]에 대한 동적 프로그래밍 모델의 반복 방정식들이다.

$$
\begin{aligned}
&H_k(0,0)=0, \text{ for all } k \in Q,\\
&H_{k^*}(j,q)=\min_{(i,j)\in A}\{[H_{k^*}(i,q')+\bar{c}_{ijk^*} \mid j\neq p(i,q'), q'+q_j=q, k^*=a(q)],\\
&\qquad [H_{k^*}'(i,q')+\bar{c}_{ijk^*} \mid j\neq p(i,q'), q'+q_j=q, k^*=a(q)]\},\\
&H_k'(j,q)=\min_{(i,j)\in A}\{[H_{k^*}(l,q')+\bar{c}_{ljk^*} \mid j\neq p(l,q'), q'+q_j=q, k^*=a(q)],\\
&\qquad [H_{k^*}'(l,q')+\bar{c}_{ljk^*} \mid j\neq p(l,q'), q'+q_j=q, k^*=a(q)]\},\\
&\text{for all } i,j,l \in V\backslash\{0\},\ i\neq l,\ q\le b_{k^*},\ k^*\in Q.
\end{aligned}
$$

SP[3] 역시 knapsack 문제를 특수한 경우로 가지고 있으므로 NP-Hard이나 그것을 풀 수 있는 pseudo-polynomial 알고리즘들이 존재한다. 그러므로 본 연구에서는 SP[1]보다 넓지만 SP[2]보다 좁은 해공간(solution space)을 가지고 있고, 또한 pseudo-polynomial 알고리즘들이 존재하는 SP[3]를 열생성 문제로 선택한다. 그리고, 그 알고리즘에 대해서는 3.2절에서 자세히 설명한다.

2.3 Set Covering Model

2.1절에서는 HVRP를 set partitioning 문제로 모형화했으나 열생성 문제인 SP[3]가 사이클을 포함하는 경로들을 생성할 수 있기 때문에 set covering 문제로 다시 모형화한다. set covering 모델은 선형완화 시켰을 때 계산상으로 set partitioning 모델에 비해 안정적이면서 branch-and-bound를 적용해서 얻은 해 또한 set partitioning 모델의 가능해가 된다.

γ_{ir}을 경로 r이 고객 i를 방문했을 때의 방문 횟수라 하고, P을 사이클을 허용하되, 2-사이클을 제외한 가능 경로들의 집합이라 하면 set covering 모델은 다음과 같다.

$$
\begin{aligned}
(\mathrm{MP}^2)\ \min\ & \sum_{r\in P} c_r x_r\\
s.t\ & \sum_{r\in P}\gamma_{ir}x_r \ge 1, \quad \text{for all } i\in V\backslash\{0\}\\
& x_r\in\{0,1\}, \quad \text{for all } r\in P.
\end{aligned}
$$

MP[2]에서 변수로 표현된 열(column)들은 가능 경로들을 의미하며, 그 수가 무수히 많으므로 MP[2]의 선형완화문제(MLP[2])를 풀기 위해 열생성 기법을 사용한다. 열은 SP[3]에 의해서 생성되며, 이렇게 생성된 열은 경로 r의 reduced cost $\bar{c}_r$이 음수일 때 추가된다. 다음은 reduced cost $\bar{c}_r$의 계산식이다.

$$\bar{c}_r = c_r - \sum_{i\in V\backslash\{\}} \pi_i^* \gamma_{ir}, \tag{3}$$

하나의 가능 경로 $r=(v_0, v_1, \ldots, v_H, v_{H+1})$, $v_0=v_{H+1}=0$일 때, 경로 r의 운송비 c_r은 $f_k+\sum_{h=0}^{H} c_{v_h v_{h+1} k}$이고, $\sum_{i\in V\backslash\{\}}\pi_i^*\gamma_{ir}$은 $\sum_{h=0}^{H}\pi_{v_h}$가 된다. 그러므로, 식 (3)은 다음과 같이 된다.

$$
\begin{aligned}
\bar{c}_r &= f_k + \sum_{h=0}^{H} c_{v_h v_{h+1} k} - \sum_{h=0}^{H}\pi_{v_h},\\
&= f_k + \sum_{h=0}^{H}(c_{v_h v_{h+1} k} - \pi_{v_h}), \qquad (4)\\
&= f_k + \sum_{h=0}^{H}(\bar{c}_{v_h v_{h+1} k}),\\
&= f_k + H_k(v_H, q) + \bar{c}_{v_H v_{H+1} k},\ \text{단}\ q=\sum_{h=1}^{H} q_{v_h}.
\end{aligned}
$$

즉, reduced cost $H_k(v_H, q)$에 차량 유형 k의 고정비 f_k와 호 (v_H, v_{H+1})의 reduced cost $\bar{c}_{v_H v_{H+1} k}$를 더해서 경로 r의 reduced cost $\bar{c}_r$을 계산한다.

여러 개의 열을 한꺼번에 추가하면 MLP[2]를 보다 빨리 풀 수 있다. 이러한 이유로 우리는 모든 노드 (i,q), $i=0,1,\ldots,N$, $q=q_i, q_i+1,\ldots,b_K$ 에서 reduced cost $\bar{c}_r$을 식 (4)와 같이 계산해서 그 값이 음수이면 모두 추가하도록 한다.

3. 알고리즘

본 절에서는 알고리즘의 전반적인 개요와 열생성 문제인 SP[3]를 풀 수 있는 psuedo-polynomial 알고리즘 중 2-사이클을 제거한 pulling 알고리즘에 대해서 설명한다.

3.1 개요

앞 절에서 언급했듯이, MLP[2]는 무수히 많은 변수를 포함하고 있기 때문에 우리는 그것을 풀기 위해 열생성 기법을 사용한다. 먼저 우리는 일부의 열을 가지고 제한된 MLP[2] (RMLP[2])를 구성하고, 필요한 열(또는 가능 경로)을 SP[3]를 풀어 reduced cost가 음수인 모든 경로를 RMLP[2]에 추가한다. 더 이상 열이 추가되지 않으면 RMLP[2]를 풀어 그 해가 정수인지 확인한다. 만약 그렇다면 우리는 HVRP의 최적 정수해를 얻게 된다. 그러나 그렇지 않을 경우에는 하나의 가능 정수해를 얻기 위해 branch-and-bound를 적용한다.

이러한 해법을 통해 얻어진 정수해는 최

적해를 보장하지는 못한다. 그러나 최적해를 구하지 못하더라도 MLP^2의 목적식 값과 branch-and-bound를 통해 얻은 정수해의 목적식 값을 비교함으로써 그 정수해의 품질을 평가해 볼 수 있다.

3.2 SP^3에 대한 알고리즘

2-사이클을 제거한 pulling 알고리즘을 설명하기 이전에 몇 가지 기호들을 정의한다. T는 V'중에서 이미 다루어진 노드들의 집합이고, $\Gamma(j,q)=\{(i,q')|(i,j)\in A, q'=q-q_j\}$는 노드 (j,q)를 선행하는 노드들의 집합이다. 다음은 알고리즘을 단계별로 설명한다.

2-사이클을 제거한 pulling 알고리즘

단계 0: **초기화(Initialization)**

$H_k(0,0)=0$, $H_k(i,q)=\infty$, $H_k'(i,q)=\infty$, $p(i,q)=$nil로 초기화한다.

단, $i=0,1,\ldots,N, q=q_i,q_i+1,\ldots,b_K$.

단계 1: **노드 선택(Node Selection)**

노드 $(i,q)\in V'\backslash T$들 중에서 q가 가장 작은 노드 하나를 선택한다. 만약 $(V'\backslash T)=\emptyset$이면 종료한다.

단계 2: **노드 처리(Node Treatment)**

$H_k(i,q)$, $H_k'(i,q)$, $k\in Q$를 최소화할 수 있는 선행 노드 $p(i,q)\in\Gamma(i,q)$를 정하고, $H_k(i,q)$와 $H_k'(i,q)$를 계산한다. $T=T\bigcup(i,q)$를 변경하고, 단계 1로 간다.

위 알고리즘의 worst case일 때의 복잡도(complexity)를 살펴보자. 단계 1에서 선택되는 노드의 총 개수는 $(N*b_K)$개이고, 단계 2에서 선택된 노드에서 고려할 수 있는 선행 노드의 수는 N개이므로, 복잡도는 $O(N^2*b_K)$가 된다.

4. 실험결과 및 분석

실험에 사용된 데이터셋은 기존에 HVRP에서 사용되던 Golden et al.[9]의 데이터를 그대로 이용하였다. 실험에 사용된 차량의 용량과 그 거리비용, 무게비용 등은 <표 1>과 같으며, b_k는 차량 유형 k의 용량이고, f_k는 고정비, g_k는 변동비를 의미한다.

<표 1> 테스트 문제로 사용된 예제들

예제 번호	지점 수	A b_k	A f_k	A g_k	B b_k	B f_k	B g_k	C b_k	C f_k	C g_k
3	20	20	20		30	35		40	50	
4	20	60	1000		80	1500		150	3000	
15	50	50	100	1.0	100	250	1.6	160	450	2.0
16	50	40	100	1.0	80	200	1.6	140	400	2.1
20	100	60	100	1.0	140	300	1.7	200	500	2.0

예제 번호	지점 수	D b_k	D f_k	D g_k	E b_k	E f_k	E g_k	F b_k	F f_k	F g_k
3	20	70	120		120	225				
4	20									
15	50									
16	50									
20	100									

우리는 이러한 예제문제들에 대해 CPLEX 7.0의 라이브러리를 사용하여 C프로그램을 작성하고, Pentium III(1GB) PC에서 실행하였다. 그 결과가 <표 2>에 정리되어 있다. <표 2>에서 'LP'는 MLP을 풀어 얻은 목적식 값이고, 'IP'는 branch-and-bound를 통해 얻은 정수해의 목적식 값이다. 'Gap(%)'은 'LP'와 'IP'를 비교한 값으로, 구하는 식은 다음과 같다.

$$Gap(\%)=\frac{LP-IP}{LP}\times 100$$

'시간(초)'는 문제를 푸는데 필요한 CPU 시간이다.

<표 2> 실험 결과

예제 번호	지점 수	best-known 값 (Value)	best-known 시간 (초)	알고리즘 수행 결과 LP	알고리즘 수행 결과 IP	알고리즘 수행 결과 Gap (%)	알고리즘 수행 결과 시간 (초)
3	20	961.03	164	951.61	961.03	0.99	5.30
4	20	6437.33	253	6369.15	6437.33	1.1	10.57
15	50	2586.37	901	2544.84	2586.37	1.63	17.99
16	50	2741.50	815	2685.92	2741.50	2.07	48.46
20	100	4047.55	1421	3995.16	4047.55	1.31	9990.17

위 표를 살펴보면 본 연구에서 제안한 알고리즘은 지금까지 좋다고 알려진 해와 동일한 결과를 나타내고 있으며, 기존의 결과들이 약 2%이내의 좋은 정수해를 제공한다는 것을 증명하였다. 시간 면에서는 기존의 발견적 해법보다 대체로 빠른 결과를 보여주고 있다.

5. 결론

본 연구에서는 HVRP의 정수계획모형을 제시하고, 열생성 기법과 branch-and-bound를 사용한 알고리즘을 개발하였다. 우리는

HVRP를 set covering 모델로 모형화하고, 그 모형의 선형완화문제(LP relaxation)를 열생성 기법을 사용하여 최적화 하였다. 선형완화문제를 풀어 얻은 해가 정수가 아닌 경우에는 하나의 가능 정수해를 얻기 위해 branch-and-bound을 적용하였다. 우리는 기존에 HVRP에 주로 사용되던 Golden et al.[9]의 데이터를 그대로 이용하여 본 연구에서 제안한 알고리즘을 테스트하였다. 그 결과 가장 좋다고 알려진 해를 대체로 빠른 시간 안에 구할 수 있었다. 또한 HVRP의 목적식 값의 하한(lower bound)를 제공함으로써 기존의 발견적 해법으로 구한 해들이 약 2%이내의 좋은 정수해를 제공한다는 것을 증명하였다.

참 고 문 헌

[1] Ballou, R., H., "Business logistics managemnet", Prentice Hall, 1998.

[2] Christofides N., A. Mingozzi and P. Toth, "State-space relaxation procedure for the computation of bounds to routing problems", Networks 11 (1981) 145-164.

[3] Desrochers M., Desrosiers J. and Solomon M., "A new optimization algorithm for the vehicle routing problem with time windows", Operations Research, Vol 40, No 2 (1992) 342-353.

[4] Desrochers M. and Verhoog J.W., "A new heuristic for the fleet size and mix vehicle routg problem", Computers and Operations Research, 18 (1991) 263-274.

[5] Gary, M. R., and D. S. Jonson, "Computer and intractability: A guide to the theroy of NP-Completeness", Freeman, San Francisco.

[6] Gendreau M. et al., "A tabu search heuristic for the heterogeneous fleet vehicle routing problem", Computers and Operations Research 26 (1999) 1153-1173.

[7] Gheysens F.G., Golden B.L, Assad A.A., "A comparison of techniques for solving the fleet size and mix vehicle routing problem", Operation Research Spectrum, 6 (1984) 207-216.

[8] Gheysens F.G., Golden B.L, Assad A.A., "A new heuristic for determining fleet size and composition",Mathematical Programming Study, 26 (1986) 233-236.

[9] Golden B., Assad A., Levy L., and Gheysens, F.G., "The fleet size and mix vehicle routing problem"Computers and Operations Research, 11 (1984) 49-66.

[10] Houck JR., D. J., J. C. Picard, M. Queyranne and R. R. Vemcganti, "The travelling Salesman Problem as a constrained shortest path problem" Theory and Computational Experience Opsearch 17 (1980) 93-109.

[11] Salhi S. and Rand G.K., "Incorporating vehicle routing into the vehicle fleet composition problem", European Journal of Operational Research, 66 (1993) 313-330.

[12] Taillard E. D., "A heuristic column generation method for heterogeneous fleet", RAIRO, (1999) forthcoming.