

차량경로문제의 경로분할모형에 관한 연구

강 성 민

A Route-Splitting Approach to the Vehicle Routing Problem

Sungmin Kang

강 성 민

경기도 부천시 역곡동 산 43-1 가톨릭대학교 경영학부

skang@catholic.ac.kr

02-2164-4285

011-359-1977

차량경로문제의 경로분할모형에 관한 연구

Abstract

차량경로문제의 Set-Partitioning 모형에 적용된 열생성 프로세스에서 차량경로를 생성하는 하위문제는 순회외판원 문제와 같은 combinatorial 구조를 가지므로 연산상의 어려움이 크다. 본 논문은 각 차량경로를 분할하여 하위문제에서 분할된 부분경로를 생성하는 경로분할모형을 소개한다. 열생성이 용이해지는 반면 주문제가 복잡해지는 단점이 있으나 이 모형은 set-partitioning 모형으로 다루기 힘든 크기의 VRP에 접근하도록 한다.

경로분할모형은 최대 199곳의 수요지를 갖는 Symmetric VRP의 실험문제에서 평균 93.5%, 수요지수 최대 70곳의 Asymmetric VRP의 실험문제에서 평균 97.6%의 하한값을 도출해 특히 Asymmetric VRP의 경우에서 잘 알려진 다른 하한값 기법들보다 우수함을 보였다. 개발된 Branch-and-Price 프로세스로는 도출된 하한값을 사용하여 수요지 최대 48곳의 Asymmetric VRP의 최적해를 구할 수 있었다.

경로분할모형은 성능이 비교되는 다른 모형과 달리 다른 크기의 차량을 다룰 수 있는 장점이 있고, Asymmetric VRP 문제에서는 현재 가장 우수한 하한값을 제시한다. 이러한 점에서 본 모형은 향후 연구가치가 있다고 판단된다.

The vehicle routing problem (VRP) is to determine a set of feasible vehicle routes, one for each vehicle, such that each customer is visited exactly once and the total distance travelled by the vehicles is minimized. A feasible route is defined as a simple circuit including the depot such that the total demand of the customers in the route does not exceed the vehicle capacity.

While there have been significant advances recently in exact solution methodology, the VRP is not a well solved problem. We find most approaches still relying on the branch and bound method. These approaches employ various methodologies to compute a lower bound on the optimal value.

We introduce a new modelling approach, termed *route-splitting*, for the VRP that allows us to address problems whose size is beyond the current computational range of set-partitioning models. The route-splitting model splits each vehicle route into segments, and results in more tractable subproblems. Lifting much of the burden of solving combinatorially hard subproblems, the route-splitting approach puts more weight on the LP master problem. Recent breakthroughs in solving LP problems (Nemhauser, 1994) bode well for our approach.

Lower bounds are computed on five symmetric VRPs with up to 199 customers, and

eight asymmetric VRPs with up to 70 customers. While it is said that the exact methods developed for asymmetric instances have in general a poor performance when applied to symmetric ones (Toth and Vigo, 2002), the route splitting approach shows a competent performance of 93.5% on average in the symmetric VRPs. For the asymmetric ones, the approach comes up with lower bounds of 97.6% on average.

The route-splitting model can deal with asymmetric cost matrices and non-identical vehicles. Given the ability of the route-splitting model to address a wider range of applications and its good performance on asymmetric instances, we find the model promising and valuable for further research.

Keywords: asymmetric vehicle routing problem, column generation, lower bound, set-partitioning model, branch-and-price

1. 도입

차량경로문제(Vehicle Routing Problem. 이하 VRP)는 전체 운행거리를 최소화하는 각 차량의 경로를 찾는 의사결정 문제이며, 제약조건은 다음과 같다.

1. N 개의 각 수요지(i)는 단일제품의 확정적인(deterministic) 주문량(q_i)을 가진다.
2. 각 차량(k)이 배송하는 수요지의 주문량의 합은 차량의 적재용량(Q_k)을 초과하지 않는다.
3. 차량 수(K)는 정해져 있으며, 각 차량은 차고(depot. $i = 0$)에서 출발하여 배송완료 후 다시 차고로 귀환한다.
4. 각 차량은 한 수요지를 1회 방문하며, 각 수요지의 주문량은 한 차량에 의해서 배송된다. 즉, 분할 배송(Split Delivery)은 허용되지 않는다.

정리하면, 모형의 첨자 및 파라미터는 다음과 같다.

$i, j =$ 차고 및 수요지. $\in \{0, 1, \dots, N\}$. $0 =$ 차고

$k =$ 차량. $\in \{1, \dots, K\}$.

모형의 투입자료는 다음과 같다.

$d_{ij} =$ 수요지 및 차고 i 에서 j 로 가는 거리.

$q_j =$ 수요지 j 의 주문량.

$Q_k =$ 차량 k 의 적재용량.

물류의 전략적 중요성이 강조되는 요즘 VRP모형의 수많은 응용사례를 볼 수 있다. 트럭, 선박, 항공기 등의 운송기관들이 전 지구적 혹은 지역적 배송망을 누비며, 제품, 부품, 고객을 나르

고 있다. 몇 가지 특이한 예를 보면 매주 정기적인 횡수의 수거가 있어야 하는 쓰레기차 경로설 정문제, 수요지 수가 많은 주류 등의 음료, 식료잡화, 우유배달의 VRP, 극히 짧은 새벽 시간대에 인쇄하는 동시에 배급해야하는 신문배급 문제 등이 있다. 본 연구와 같은 순수 VRP의 이론적 연구는 현실의 배송시스템의 구축에 관한 연구에 기여할 것으로 사료된다.

VRP는 그 동안 대개 휴리스틱으로 접근되었으며, 최적화는 실용적이 아닌 것으로 여겨졌지만, 이제 견해를 바꾸어야 할 몇 가지 이유가 있다 (Fisher, 1994). 첫째, 컴퓨팅 환경의 발달로 최적해 탐색의 시간과 비용이 크게 내려간 점, 둘째, 거리 및 네트워크 자료의 정확성이 현저히 개선된 점, 셋째로, 최적화 이론이 발달한 점 등을 들 수 있다. 최적화 알고리즘이 시간상 최적해 도출 전 중단되더라도 최적해와의 차이에 대한 정보를 제공한다는 점, 최적화기반의 휴리스틱 (optimization-based heuristic)의 가능성 등은 최적화 접근법의 장점이다.

VRP의 가장 고전적인 모형 중 하나인 Set-Partitioning Model에서는 변수가 위의 제약조건을 충족하는 유효한 한 차량경로를 의미하며, 따라서 K 개의 변수가 한 해를 구성한다. 경로선택과 정에서 VRP with time windows 등 현실의 다양한 제약을 모형화할 수 있는 유연함이 강점이다. 또한 이 모형은 LP relaxation에 의한 하한값(lower bound)이 매우 강한(tight) 것으로 알려져 있다. 모형의 설명을 위해 잠시 각 차량의 적재용량이 같다고(Q) 가정한다. 이 가정은 경로 분할 모형에서는 불필요하다. 또한 본 절에 한해 다음의 기호를 정의한다.

R = 유효한 경로의 집합.

a_{ir} = 수요지 i 가 경로 $r (\in R)$ 에 있으면 1, 아니면 0.

d_r = 경로 r 의 총 차량이동거리. 경로의 arc들 거리의 합.

의사결정변수 x_r 이 경로 r 이 선택되면 1, 아닌 경우에는 0의 값을 갖는다면, Set-Partitioning 모형은:

$$(VRP-SP) \text{ minimize } \sum_{r \in R} d_r x_r$$

subject to

$$\sum_{r \in R} a_{ir} x_r = 1 \quad \forall i = 1, \dots, N \quad (1)$$

$$\sum_{r \in R} x_r = K \quad (2)$$

$$x_r \in \{0, 1\} \quad \forall r$$

이다. 제약식 (1)은 각 수요지가 정확히 한 차량에 의해 1회 방문되어야 한다는 제약을 나타내고, 제약식 (2)는 각 차량에 하나씩 K 개의 경로를 선택한다.

VRP에는 수요지 i 에서 j 까지 거리가 j 에서 i 까지 거리와 같다고 가정하는 대칭모형 (Symmetric VRP, 이하 SVRP)과 서로 다른 거리를 허용하는 비대칭 모형 (Asymmetric VRP,

이하 AVRP)이 있다. 기존의 주요한 연구는 대개 SVRP를 대상으로 하는데, SVRP를 위해 개발된 모형이나 해법은 물론 AVRP에는 적용되지 않는다. 반면에 SVRP가 AVRP의 특수한 경우지만, AVRP의 해법 알고리즘은 SVRP에 적용될 때 일반적으로 열등한 성능을 보인다(Toth와 Vigo, 2002).

2. 문헌 조사

VRP는 유명한 TSP(Traveling Salesman Problem)의 일반적(generalized) 형태로서 이론적으로 매우 흥미로운 연구주제이며 현실적으로도 응용성이 높은 중요한 문제이다. 그러나 VRP는 같은 규모(N)의 TSP에 비해 훨씬 어려운 문제이다. VRP의 여러 모형에 대해서는 Bodin 등(1983), Laporte와 Nobert(1987), Laporte(1992)와 Fisher(1992), 그리고 최근의 Toth와 Vigo(2002) 등에서 찾아볼 수 있다. 최근 최적화 연구에 많은 진전이 있었지만 NP-Hard 문제인 VRP는 well-solved problem이 아니며, 여전히 분지한계법(Branch and Bound)에 크게 의존한다. 분지한계법의 효능은 하한값에 의해 좌우되므로 최적값에 근사(tight)한 하한값은 매우 중요하며, 지난 20여년간 우수한 하한값을 도출하는 방법론에 관한 다양한 시도가 있었다.

Christofides, Mingozzi와 Toth(1981)는 minimum K -degree center tree(차고 노드의 degree가 K 인 shortest spanning tree)를 통한 하한값과 minimum q -route(수요지 주문량의 합이 q 인 경로)를 사용한 하한값을 소개하며, 수요지 25곳 차량 8대의 SVRP 문제를 최적화하였다.

SVRP 대상의 b -matching 모형의 최적해는 각 수요지 노드의 degree가 2이며 차고노드의 degree는 $2K$ 인 최소비용 cycle의 집합이다. 모든 cycle이 차고를 통과하며, 차량의 적재용량을 지키는 경우에는 VRP의 최적해이며, 아닌 경우는 하한값을 제공한다. Miller(1995)는 b -matching에 의한 하한값을 subtour제거 제약식과 차량적재용량 제약식의 Lagrange 쌍대화를 통해 강화했다.

Fisher(1994)는 TSP의 해법으로 잘 알려진 Held와 Karp(1970)의 1-tree relaxation을 일반화시킨 K -tree relaxation을 SVRP에 적용하였으며, 차량적재용량 제약식과 노드 degree 제약식의 Lagrange 쌍대화를 통해 강화한 하한값을 제시하였다.

SVRP의 b -matching에 대응하는 AVRP의 하한값은 Laporte, Mercure, and Nobert(1986)가 제시한 AP(assignment problem) relaxation이다. AP 문제는 $O(n^3)$ 시간의 효율적인 코드가 공개되어 있으나(Dell'Amico와 Toth, 2000) 하한값은 약한 편이다.

또 하나의 잘 알려진 AVRP 모형의 하한값인 KSSA(K -shortest spanning arborescence) 하한값은 $O(n^2)$ 로 빠르게 얻는 장점이 있는 반면, 일반적으로 약하며 AP 하한값에 비해서도 열등하다. KSSA 하한값이 분지한계법에 적용된 예는 아직 없다(Toth와 Vigo, 2002).

Fischetti, Toth와 Vigo(1994)는 병립이 불가능한 arc집합을 이용한 disjunctive 하한값, min-cost flow 모형을 이용한 하한값, AP 하한값 등 서로 다른 하한값을 더해 나가며 하한값을

강화하는 ADD(additive bounding) 기법을 제시하였다. 따라서 개별 하한값인 AP 하한값에 비해서 우수하다. 예를 들어 AVRP 실험문제에서 AP가 평균 91.3%인데 반해 94.6%의 최적값대비 백분율을 보였다.

AVRP의 Set-Partitioning 모형(VRP-SP)에서 변수는 하나의 경로이다. 경로가 되는 수요지의 조합은 거의 무한히 많기 때문에 모형에 모든 변수를 다 포함하는 것은 현실적으로 불가능하다. 따라서 VRP-SP의 해법은 대개 열생성(column generation)기법을 채택한다. 그러나 열생성 기법은 열, 즉 변수의 생성이 Combinatorial 문제로서 난이도가 문제 크기에 따라 급격하게 증가하는 과제라는 데 그 어려운 점이 있다. 한 변수의 비용(목적함수 계수)은 그 차량경로의 총 운행거리인데 이는 경로에 포함된 수요지 수와 같은 규모의 TSP 문제를 푸는 것과 같다. 열생성 기법에서는 이와 같은 경로를 수없이 계산하고 생성해야 하므로 그 생성 속도에 성능의 관건이 있다. 그러므로 지금까지 VRP-SP의 성공적인 적용사례는 문제 규모가 작아 경로생성에 큰 어려움이 없거나, 혹은 부수적인 제약조건이 있어 유효한 경로의 수가 현저히 작은 경우에 국한된다. 전자의 예로서 Agarwal, Mathur와 Salkin(1989)이 수요지 25곳의 VRP를 최적화하였고, 후자의 예로서 Desrochers, Desrosiers와 Solomon(1992)의 매우 좁은 시간대를 갖는 VRP with time windows의 연구가 있다. Set Partitioning 모형의 경우는 아니지만 경로를 다루는 q-route relaxation(Christofides 등, 1981)도 경로의 평균 수요지가 3.8곳인 경로생성이 쉬운 VRP를 취급했음을 알 수 있다.

본 연구는 Set-Partitioning 모형의 어려움을 경감하는 새로운 모형을 소개한다. 본 절에서 소개한 기존연구의 모형은 모두 차량의 적재용량이 같다고(homogenous vehicles) 가정한다. 경로 분할모형은 Asymmetric VRP 모형이며, heterogeneous vehicle을 허용한다.

3. 경로분할 모형

경로분할모형은 VRP-SP모형을 기반으로 한다. VRP-SP에서는 변수가 하나의 차량경로이지만, 경로분할모형(Route-Splitting Model)에서는 변수가 경로를 몇 부분으로 나눈 분할경로이기에 생성하기가 훨씬 용이하다. 경로의 생성은 경로의 최대 길이가 짧을수록 용이하기 때문이다. 반복적으로 변수를 생성해야 된다는 점에서 변수생성 속도는 매우 중요하다. 분할경로모형은 이와 같은 조건을 얻는 것이다. 한편 경로분할모형은 분할경로들을 각 차량경로로 결합시키는 추가 제약조건을 필요로 한다. 그 결과는 경로를 분할하지 않는 모형에 비해 복잡해지고 규모가 커진 주문제이다. 따라서 경로분할모형은 변수를 생성하는 하위문제의 부담을 줄이는 대신 주문제의 부담을 늘리게 되는데 이것은 시도해 볼 가치가 있다. 왜냐하면 LP모형인 주문제는 문제의 복잡성이나 크기의 영향을 적게 받기 때문이다.

임의의 분할경로 r 은 차고와 수요지들의 한 배열 순서를 나타낸다. 경로분할모형의 기호를 다음과 같이 정의한다:

\mathbb{R} = 모든 유효한 분할경로의 집합. $r \in \mathbb{R}$.

R = 분할경로의 부분집합. $R \subset \mathbb{R}$.

$R(k) =$ 차량 k 의 분할경로의 집합. $\subset R$.

$R(k, s) =$ 차량 k 의 s 번째 분할경로의 집합. $\subset R(k)$.

$s(k) =$ 차량 k 의 경로의 분할경로의 수.

$s =$ 분할경로의 순번. $s = 0, 1, \dots, s(k)$.

$B(r) = r$ 이 O형, 혹은 C형인 경우에는 r 에 포함된 모든 수요지의 집합. r 이 A형, 혹은 B형인 경우에는 r 의 최종 수요지를 제외한 수요지의 집합.

$b(r) = r$ 의 최초 수요지.

$e(r) = r$ 의 최종 수요지.

$q_r =$ 분할경로 r 의 주문량. $q_r = \sum_{i \in B(r)} q_i$

$d_r =$ 분할경로 r 의 차량주행거리. r 에 속한 arc들 거리의 합.

분할경로는 4가지 형태가 있다. 첫째 O형 분할경로는 차고에서 출발하여 차고로 돌아오는 분할되지 않은 하나의 경로 자체이다. 작은 경로는 분할하지 않는 것이 효율적이다. A형 분할경로는 차고에서 출발하여 한 수요지에서 끝나는 형태이며, B형 분할경로는 한 수요지에서 다른 수요지까지의 차고가 포함되지 않은 구간이다. 마지막으로 C형 분할경로는 한 수요지에서 출발하여 차고에서 끝나는 형태의 구간이다. 특히 B형의 분할경로가 단 1개의 수요지를 포함하는 경우, 단일노드의 형태가 된다. O형을 제외하면, 경로는 2개 이상의 분할경로의 연결로 구성되는데, 차량 k 의 경로의 분할경로의 수가 $s(k)$ 라고 할 때, 경로는 $s=1$ 에서 $s(k)$ 까지의 분할경로의 연결이다. 이때 $s=1$ 의 분할경로는 A형이고, $s=s(k)$ 의 분할경로는 C형이며, 그 사이의 분할경로는 ($s(k) > 2$ 인 경우) 모두 B형이다.

<그림 1>은 차량 2대로 10개의 수요지를 방문하는 VRP해의 한 예를 보이고 있다. 그림에서 $r_1=(0,1,2,3)$, $r_2=(3,4,5)$, $r_3=(5,6,7,8,0)$, 그리고 $r_4=(0,9,10,0)$ 라면, r_1 , r_2 , r_3 는 각각 A형, B형, C형의 분할경로이며, r_4 는 O형이다. $s(k) = 1$ 인 차량 2의 분할경로는

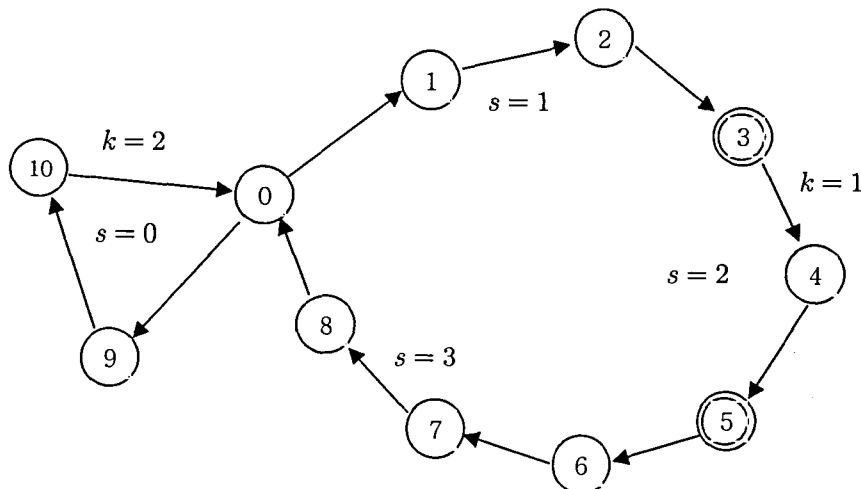


그림 1 : 수요지 10곳, 차량 2대의 VRP의 해의 예

$s=0$ 인 0형의 r_4 이다. 차량 1의 경우 $s(k)=3$ 이며, $s=1, s=2, s=3$ 의 분할경로는 각각 r_1, r_2, r_3 이다. 또한 $B(r_1) = (1,2)$ 이고 $e(r_1) = 3$ 이다. $b(r_2) = 3$ 이고, $B(r_2) = (3,4)$ 이며, $e(r_2) = 5$ 이다. $b(r_3) = 5$ 이고, $B(r_3) = (5,6,7,8)$ 이며, $B(r_4) = (9,10)$ 이다. 한편 <그림 1>에서 수요지 3과 수요지 5에서는 연이은 두 분할경로가 연결되고 있다. 이와 같이 두 분할경로의 연결지점 역할을 하는 수요지의 주문량은 뒤의 분할경로에 할당한다. 즉, 예를 들어, 수요지 3의 주문량은 분할경로 r_2 에 포함된다.

모형의 이진변수 x_r 은 분할경로 r 이 차량경로의 부분구간으로 선택되면 1, 아니면 0의 값을 가진다. 따라서 변수 x_r 과 분할경로 r 은 이후의 설명에서 같은 의미로 쓰인다. 모형에 포함된 분할경로의 집합이 R 일 때 경로분할모형은

$$(RS(R)) \text{ minimize } \sum_{r \in R} d_r x_r$$

subject to

$$\sum_{r: i \in B(r)} x_r = 1 \quad \forall i = 1, \dots, N \quad (3)$$

$$\sum_{r \in R(k)} q_r x_r \leq Q_k \quad \forall k = 1, \dots, K \quad (4)$$

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,1)} x_r = 1 \quad \forall k = 1, \dots, K \quad (5)$$

$$\sum_{r \in R(k,s): b(r)=i} x_r - \sum_{r \in R(k,s-1): e(r)=i} x_r = 0 \quad \forall i \forall k, s = 2, \dots, s(k) \quad (6)$$

$$x_r \in \{0,1\} \quad \forall r \in R.$$

모형 $RS(R)$ 에서 $R = \mathbb{R}$ 인 경우, 즉, RS 의 최적해는 $VRP-SP$ 의 최적해와 일치한다. $RS(R)$ 의 분할경로는 결합되어 경로를 형성하기 위해 다음과 같은 조건을 만족시켜야 한다:

- 가) 각 수요지는 한 분할경로의 $B(r)$ 에 1회 포함된다.
- 나) 각 차량에 하나씩 K 개의 그룹을 형성하며, 각 그룹은 하나의 0형 분할경로, 혹은 순서대로 연결되는 $s(k)$ 개의 분할경로로 구성된다.
- 다) 각 그룹의 수요지의 주문량의 총 합이 해당 차량의 적재용량을 초과하지 않는다.

모형의 목적함수식은 차량 총 이동거리를 최소화하는 것이다. 제약식 (3)은 가)의 조건에 해당하며, 식 (4)는 다)의 조건을 제약한다. 제약식 (5)와 (6)은 $s(k) = 1$ 인 경우 0형 분할경로를 선택하며, $s(k) \geq 2$ 인 경우 각 차량의 $s(k)$ 개의 분할경로가 잇달아 연결되어 경로를 형성하도록 $s \geq 2$ 의 분할경로의 최초 수요지가 바로 직전의 분할경로의 최종 수요지와 같도록 제약

하여 나)의 조건이 만족하도록 한다.

4. 알고리즘

기호 $z(\cdot)$ 은 모형 (\cdot) 의 최적값을 표시한다. 그 외 기호를 정의하면:

$n(r) =$ 분할경로 r 에 포함된 수요지의 수.

$B = n(r)$ 의 최대 크기를 나타내는 파라미터. 즉, $1 \leq n(r) \leq B$.

본 연구의 하한값 RS1은 모형 RS의 LP relaxation인 LRS의 최적값 $z(\text{LRS})$ 이다. LRS는 수많은 변수를 포함하여 모두 표시하기는 현실적으로 불가능하므로, 필요한 x 변수를 반복적으로 추가하는 열생성 프로세스를 채택한다. 열생성의 한 회차에서는 분할경로의 현재의 부분집합 R ($\subset \mathbb{R}$)을 포함한 $\text{LRS}(R)$ 의 최적 쌍대변수값을 통해서 하한값을 더욱 내릴 수 있는 변수를 하위문제에서 찾는다. 다음 회차에서는 생성된 변수를 추가한 모형 $\text{LRS}(R')$ 를 최적화한다. 이 과정을 반복하며, 더 이상 reduced cost가 음수인 변수가 없으면 중단한다. 이때 $\text{LRS}(R)$ 은 $\text{LRS}(\mathbb{R})$, 즉 LRS와 최적해가 일치한다.

열생성과정은 주문제와 하위문제를 반복적으로 수행한다. 이때 주문제의 하한값은 $s(k)$ 의, 하위문제의 속도는 B 의 영향을 받는다. 극단적으로 모든 k 에 대해 $s(k) = 1$ 이면 경로분할모형은 set-partitioning 모형과 일치하므로 하한값이 tight할 것이다. 이때 B 는 최적 차량경로 전체를 나타낼 수 있도록 충분히 커야 한다. 정리하면 $s(k)$ 가 작을수록 하한값이 강해지는 반면 B 는 커져야 한다. 문제는 하위문제에서의 열생성 시간이 B 의 크기에 따라 급격히 상승한다는 것이다. 본 연구에서는 파라미터 B 의 값을 (문제에 따라 조정하지 않고) 모든 문제에서 일괄적으로 $B = 5$ 의 고정값을 사용하였다. 사전실험에서는 $B = 4$ 에서 6 으로 B 의 값이 한 단위 올라갈 때 하위문제의 풀이시간은 큰 폭으로 상승하였으며, 하한값은 평균 0.5%정도 상승하는 결과를 보였다.

본 연구에서는 모형에서 각 차량 k 의 경로의 분할경로의 수를 나타내는 $s(k)$ 는 B 에 따라 사전에 결정하였다. 부록 A의 나)는 $s(k)$ 의 도출 공식을 보이고 있다. 차량 k 가 방문하는 최대 수요지수를 (문제의 데이터에 따른) $U(k)$ 로 표시할 때, $s(k) = \left\lceil \frac{U(k) - B}{B - 1} \right\rceil + 1$ 로 구한다. 예를 들어, $U(k) = 17$ 이면, $s(k)$ 는 4가 된다. $s(k)$ 에 따라 각 분할경로 길이의 상한과 하한 $(L(k, s), U(k, s))$ 이 결정된다.

$\text{LRS}(R)$ 의 식 (3), (4), (5), (6)에 해당하는 쌍대변수가 각각 $\beta_i, \gamma_k, \alpha_k$ 와 π_{isk} 라면, 변수 x_r 의 Reduced Cost는 변수 x_r 이 차량 k 의 s 번째 분할 경로라고 할 때 다음과 같다:

$$s = 0 \text{ 일때} \quad d_r - \sum_{i \in B(r)} \beta_i - q_r \gamma_k - \alpha_k$$

$$\begin{aligned}
s = 1 \text{ 일때} & \quad d_r - \sum_{i \in B(r)} \beta_i - q_r \gamma_k - \alpha_k + \pi_{e(r),s+1,k} \\
1 < s < n(k) & \quad d_r - \sum_{i \in B(r)} \beta_i - q_r \gamma_k - \pi_{b(r),sk} + \pi_{e(r),s+1,k} \\
s = n(k) & \quad d_r - \sum_{i \in B(r)} \beta_i - q_r \gamma_k - \pi_{b(r),sk}
\end{aligned}$$

열생성 중 현재 회차의 LRS(R)에서,

$$RS1 = \sum_k \min (z_{0k}, \sum_{s=1}^{s(k)} z_{sk}) + z(LRS(R)) \quad (7)$$

는 $z(LRS)$ 의 하한값을 형성한다 (Lasdon, 1970). 부록 B는 식 (7)의 도출과정을 설명한다. z_{sk} 는 각각 차량 k 의 s 번째 분할경로에 해당하는 변수 중 최소의 reduced cost 값을 의미한다. RS1은 열생성 프로세스에서 단조적으로(monotonously) 증가하지는 않지만, $z(LRS)$ 에 수렴한다.

VRP-SP모형은 LP relaxation 하한값이 매우 강한 것으로 알려져 있다. 예를 들어 Agarwal 등(1989)의 경우 평균 하한값은 $N=20$ 정도의 문제에서 98.5%였다. 경로분할모형의 하한값은 VRP-SP모형의 하한값을 능가할 수는 없다. 부록 A의 다)는 본 연구에서 사용된 하위문제의 해법알고리즘 GenColumn을 소개하고 있다.

5. 실험 연산

실험연산에는 800MHz Pentium PC가 사용되었으며, 분지한계 열생성(Branch and Price) 알고리즘은 Microsoft C++ 2.0으로 프로그램했으며, LP 주문제는 CPLEX Callable Library 6.0을 사용해 최적화하였다.

본 절에서는 SVRP와 AVRP의 대표적 하한값과 RS모형의 RS1과 RS2를 실험하여 비교한다. SVRP의 대표적 하한값으로는 b-matching, K -tree, AVRP의 대표적 하한값으로는 KSSA, AP, ADD 등이 선택되었다.

5.1 Symmetric VRP의 하한값 계산

NP-hard 문제에 적용된 알고리즘의 최적화 수행도는 데이터에 따라 큰 편차를 보인다. VRP도 예외가 아니며 차량을 추가하거나 수요자의 주문량을 일부 변화시키는 것만으로 전혀 다른 결과를 얻을 수 있다. 따라서 적절한 비교를 위해서는 잘 알려진 테스트 문제를 사

용하는 것이 요구된다. 본 연구에서는 VRPLIB의 테스트 문제(Vigo, 2000)를 사용하며, 특히 SVRP의 실험연산에는 Fisher(1994)와 같은 문제를 사용하여 비교한다. 테스트문제 중 E50, E75와 E100은 Eilon 등(1971)에 소개된 자료이며, E150은 E50의 수요지와 E100의 수요지를 합치고, 차고의 위치와 차량의 적재용량은 E100을 사용한 것이다. E199는 E150에 E75의 첫 49곳의 수요지를 합한 것이다. 각 문제의 d_{ij} 는 실수(real number)로 계산되었으며, 차량은 동일하다.

ID	N	K	UB	RS1		RS2		
				하한값	CPU	intK	하한값	CPU(분)
E50	50	5	524.6*	503.2	.2	2	507.7	1.4
						3	508.6	1.7
						4	509.6	3.2
						5	509.6	2.6
E75	75	10	835.3	769.5	.8	2	784.0	7.7
						3	787.5	9.0
						4	790.2	13.4
						10	796.6	85.5
E100	100	8	826.1	771.2	7.0	2	783.3	35.0
						3	786.0	61.5
						4	790.9	170.8
						8	795.1	832.4
E150	150	12	1028.4	924.4	35.3	3	944.7	519.1
						4	948.2	727.0
E199	199	16	1334.6	1155.6	112.3	3	1168.5	763.2
						4	1171.1	1055.0

표 1: Symmetric VRP에 적용된 경로분할모형의 결과 (*최적해)

<표 1>은 실험연산의 결과를 보여준다. N과 K는 각각 문제의 수요지수와 차량수이며, UB는 알려진 가장 우수한 해의 값이다 (Fisher, 1994). RS1과 RS2 열은 하한값과 사용된 CPU 시간(분 표시)을 표시한다. 하한값 RS1은 RS 모형의 LP 최적값이고, 하한값 RS2는 RS1을 다음과 같은 방법으로 강화한 것이다: 먼저 $n(k)$ 가 차량 k 의 경로에 포함된 수요지의 수라고 하자. RS의 최적해에서 각 차량경로의 $n(k)$ 는 정수다. 따라서 LRS의 각 차량경로의 $n(k)$ 의 정수화를 통해 RS1을 강화할 수 있다. 정수화는 분지한계 열생성법을 사용했다. 이 기법은 분지한계나무의 각 노드에서 열생성을 수행하는 것이다. 즉, 각 노드의 LRS모형에 $n(k)$ 의 크기를 정수로 제약하는 제약조건이 추가되므로 열생성 프로세스를 통해서 다시 최적화해야 한다 (본 연구에서는 분수인 $n(k)$ 를 대상으로 $n(k) \leq \lfloor n(k) \rfloor$ 이거나 $n(k) \geq \lfloor n(k) \rfloor + 1$ 인 두 갈래의 branching을 사용함). 파라미터 intK는:

$$\text{intK} = n(k) \text{가 정수로 제약되는 차량의 수}$$

로 가장 강화된 RS2는 $\text{intK} = K$ 일때 얻어진다. <표 1>에서 $\text{intK}=2$ 정도의 작은 크기에도 RS1에 비해 RS2가 하한값이 크게 상승하나 그 증가폭은 추가적인 intK의 증가에는 감소한다.

E50, E75와 E100에서 $\text{int}K = K$ 인 가장 강화된 RS2 평균은 96.3%이나, 이와 같은 하한값을 얻기 위해서는 큰 CPU시간이 필요하다는 것을 알 수 있다.

<표 2>에서 KSSA, AP, ADD는 AVRP 모형의 하한값이고, b-matching, K-tree는 SVRP 모형의 하한값이다. RS1과 RS2를 포함해 표시된 값들은 UB의 백분율로 표시되고 있다. 적절한 속도와 일관성있는 비교를 위해 RS2는 모두 $\text{int}K=4$ 를 사용한 하한값을 표시하였다.

	(%)KSSA	(%)AP	(%)ADD	(%)b	(%)K-tree	(%)Fisher	(%)RS1	(%)RS2
E50	79.4	80.9	87.5	87.9	84.9	96.7	95.9	97.1
E75	69.2	71.0	76.1	76.7	76.2	90.5	92.1	94.6
E100	77.5	80.7	86.1	86.4	81.5	95.1	93.3	95.7
E150	73.6	68.6	77.6	80.5	78.4	90.7	89.9	92.2
E199	64.4	62.7	70.0	70.2	71.9	82.2	86.6	87.9
	72.8	72.8	79.5	80.3	78.6	91.0	91.6	93.5

표 2: UB의 백분율로 표시된 잘 알려진 하한값들과 RS 하한값의 비교

SVRP는 AVRP의 특수한 경우지만, AVRP용 하한값기법은 SVRP에서 좋은 결과를 보이지 못한다(Toth 와 Vigo, 2002). <표 2>를 보면 AVRP용인 KSSA, AP, ADD가 b-matching, K-tree에 비해 열등한 결과를 보이고 있다. 반면 본 연구의 RS1 모형은 AVRP용이지만 SVRP에서 경쟁력이 있다는 것을 보이고 있다. 그러나 위의 하한값 중 어느 것도 분지한계법에 사용하기에는 충분히 강하지 않다. 대부분 UB대비 백분율이 90%도 넘지 못하기 때문에 분지한계 프로세스에서 거의 쓸모가 없다. Fisher(1994)는 K-tree 하한값을 노드 degree 제약식과 차량용량 제약식의 Lagrange 쌍대화로 강화하여 우수한 하한값을 구하였다. K-tree 하한값의 UB대비 평균 백분율이 78.6%인데 반해 Lagrange법으로 강화된 하한값의 UB대비 평균 백분율은 91%이다. RS모형의 RS1은 Fisher의 강화된 Lagrange 하한값과 견줄만하며, RS2는 평균 93.5%의 우수한 UB대비 백분율을 보인다. Fisher의 Lagrange 하한값과 RS모형의 하한값의 주요한 특성을 비교하면:

1. Fisher는 SVRP 모형, RS는 AVRP 모형
2. 둘 다 d_{ij} 를 실수(real number)로 취급한다.
3. Fisher는 동일한 차량을 가정. RS는 서로 다른 차량을 허용.
4. Fisher(1994)가 사용한 컴퓨터는 Apollo Domain 3000 (0.071 Mflops)는 본 연구에서 사용된 Pentium III 800MHz (140 Mflops)에 비해 약 2000배 정도 속도차이가 난다. Fisher가 E50에서 96분을 사용하고, RS1은 약 0.2분을 사용하므로, E50에서 Fisher와 RS1의 속도를 비교하면 약 5배정도 Fisher가 빠르다는 것을 알 수 있다. 또한 RS는 규모가 커짐에 따라 속도가 느려지기 때문에 E199에 있어서는 Fisher CPU 1186분, RS1 CPU 112분으로 약 190배의 속도 차이가 난다.
5. Fisher는 Single-Customer Route(수요지가 하나인 경로)를 허용하지 않는다. RS는

그러한 제약이 없다. 따라서 정확한 비교를 위해서는 Fisher의 하한값은 약간 하향 조정해야 한다 (Toth 와 Vigo, 2002).

5.2 Asymmetric VRP의 하한값 계산

AVRP의 실험연산에는 Vigo(1999)의 테스트 문제가 사용되었다. 이 실험문제군은 최대 70곳의 수요지와 차량 3대의 VRP를 포함한다. <표 3>에 제시된 UB는 최적값이다 (Toth 와 Vigo, 2002). RS2는 vigo34의 경우 intK=2, 나머지 문제들은 intK=3을 사용하였다.

ID	N	K	UB	RS1		RS2	
				하한값	CPU(분)	하한값	CPU(분)
vigo34	33	2	1406*	1330.1	.1	1349.8	.2
vigo36	35	3	1644*	1565.8	.1	1585.1	.6
vigo39	38	3	1654*	1626.7	.1	1647.5	.8
vigo45	44	3	1740*	1695.5	.3	1734.0	2.7
vigo48	47	3	1891*	1822.9	.3	1862.9	6.1
vigo56	55	3	1739*	1623.7	.5	1673.4	9.4
vigo65	64	3	1974*	1905.9	1.2	1930.8	17.2
vigo71	70	3	2054*	1945.0	6.1	1981.9	38.3

표 3: Asymmetric VRP의 RS 하한값 (*최적해)

<표4>는 AVRP에 관한 잘 알려진 하한값 기법들을 Vigo 문제에 적용시킨 결과이다. KSSA와 AP는 각각 최적값 대비 평균 77.1%와 91.3%로 열등하다는 것을 알 수 있다. ADD는 평균 94.6%이다. 반면 RS1은 95.3%, RS2는 97.6%로 우수함을 보인다.

ID	(%)KSSA	(%)AP	(%)ADD	(%)RS1	(%)RS2
vigo34	78.7	85.8	90.1	94.6	96.0
vigo36	75.2	90.9	93.2	95.2	96.4
vigo39	77.6	93.8	96.7	98.3	99.6
vigo45	75.6	93.4	95.7	97.4	99.7
vigo48	79.0	93.6	97.2	96.4	98.5
vigo56	75.4	88.5	94.3	93.4	96.2
vigo65	75.6	92.6	95.5	96.6	97.8
vigo71	79.3	91.7	94.6	94.7	96.5
	77.1	91.3	94.6	95.8	97.6

표 4: Asymmetric VRP에 적용된 여러 하한값의 최적값에 대한 백분율

5.3 최적해 도출을 위한 분지한계 열생성 프로세스

LRS의 최적값은 정수 최적값의 하한값이 된다. 열생성을 통한 LRS의 최적화과정에서 모형에

포함된 변수는 전체 변수의 부분이기에 그 변수만을 대상으로 분지한계 탐색을 하면 최적해를 얻는다는 보장이 없다. 따라서 분지한계 열생성 기법이 필요하다. 이 기법은 분지한계 탐색나무의 각 노드에서 열생성을 전개하는 것으로 정수 최적해를 얻는다.

분할경로나 경로와 같이 복수의 arc를 포함한 변수를 분지한계과정에서 branching하는 것은 쉽지 않다. 변수값=1로의 고정은 단순하지만, 0으로의 고정은 매우 어렵기 때문이다. 이러한 점을 인식한 Desrochers 등(1992)은 VRP with time windows의 연구에서 경로 변수가 아닌 arc 변수 x_{ij} 를 사용하였다. $x_{ij}=1$ 로 고정하기 위해서는 $d_{ij}=-\infty$, $x_{ij}=0$ 으로 고정시키기 위해서는 $d_{ij}=\infty$ 로 변경한다. 본 연구의 분지한계 열생성 프로세스에서도 arc값 고정을 하는데, arc값을 0으로 고정하기 위해서는 $d_{ij}=\infty$ 를 사용하지만, arc값을 1로 고정하는 것은 노드결합을 한다. 경로분할모형은 arc에 해당하는 x_{ij} 형의 변수는 사용하지 않기 때문에 arc값은 해당 arc를 포함하는 모든 분할경로의 변수값을 더해서 얻는다. arc값이 0.5에 가까운 arc를 branching 대상으로 선택하였다.

노드결합 조작

1. 현 노드의 집합을 N 이라고 한다.
2. 차고가 아닌 수요지 i 와 j 를 노드 h 로 결합한다 (차고 노드는 결합하지 않는다).
3. $N' = N - i - j + h$
4. $d_{mh} = d_{mi}$ for all $m \in N' - h$
5. $d_{hn} = d_{jn}$ for all $n \in N' - h$.

이미 노드결합된 노드도 노드결합의 대상이기 때문에 여러 개의 노드가 결합된 노드가 있을 수 있다. 이 기법의 장점은 분지한계 열생성 프로세스에서 고정되는 변수가 많아지면서 문제가 점점 쉬워진다는 것이다. 노드결합 기법은 SVRP의 경우에도 변경된 그래프의 d_{ij} 는 Asymmetric이 되기 때문에 Asymmetric 모형에만 적용가능하다.

ID	#branching nodes	(%)RS2	CPU(분)
vigo34	815	96.0	15.7
vigo36	6981	96.4	145.9
vigo39	25	99.6	3.3
vigo45	15	99.7	13.2
vigo48	5173	98.5	474.6

표 5: 분지한계 열생성 프로세스의 종료까지의 결과

<표 5>는 분지한계 열생성 프로세스가 종료할 때까지 총 노드수 및 사용시간을 표시하고 있다. 하한값이 99%이상인 vigo39 와 vigo45 에서는 30개 이하의 노드로 전체 프로세스가 최적해를 증명하며 종료되었지만, 하한값 비율이 평균 97%인 vigo56 이상 크기의 문제는 1000분의 CPU 시간동안 최적해에 도달하지 못하였다.

6. 맺음말

본 연구는 Set-Partitioning 모형을 기반으로 한 VRP의 새로운 모형을 소개하고, 최적화를 시도한다. VRP는 잘 알려진 TSP의 generalization으로서 이론적으로나 현실적으로 중요한 문제이다. VRP의 최적화에서 최적값에 근사한 우수한 하한값의 도출은 필수불가결하여 그간 방법론에 관한 다양한 시도가 있었다.

분할경로모형은 차량별 경로를 작은 길이로 분할해서 변수를 빠르게 생성한다. 그 대가는 경로를 분할하지 않는 모형에 비해 복잡해지고 규모가 커진 주문제이다. 그러나 주문제는 LP 모형이기 때문에 복잡성이나 규모의 영향을 적게 받는다는데 경로분할 접근법의 유용성이 있다. 특히 최근의 LP 해법의 비약적인 발전은 이러한 시도에 타당성을 부여한다 (Nemhauser, 1994).

Symmetric VRP는 Asymmetric VRP의 특수한 경우지만, Asymmetric VRP용 하한값 기법은 Symmetric VRP에서 좋은 결과를 보이지 못한다고 알려져 있다(Toth 와 Vigo, 2002). 경로분할모형은 Symmetric VRP의 실험문제에서 평균 93.5%의 하한값을 제시해 Symmetric VRP에서 경쟁력이 있다는 것을 보이고 있다. 또한 Vigo(1999)의 테스트 문제를 사용한 Asymmetric VRP의 실험연산에서 경로분할모형은 평균 97.6%의 하한값을 도출해 잘 알려진 다른 하한값 기법들보다 우수함을 보인다.

분지한계 열생성(branch-and-price) 프로세스는 본 연구에서 하한값과 정수최적해 도출을 위해 사용되는데, 이 기법은 분지한계 탐색나무의 각 노드에서 열생성을 전개하는 것으로, Toth 와 Vigo (2002)는 분지한계 열생성에 관한 연구가 "still almost uncovered" 라고 지적한다. 본 연구의 분지한계 열생성 프로세스에서는 arc값을 1로 고정하기 위해 노드결합을 한다. 이 기법의 장점은 고정되는 변수가 많아지면서 문제가 점점 쉬워진다는 것이다. 노드결합 기법은 경로분할모형과 같은 Asymmetric 모형에서만 사용 가능하며, 본 연구자가 알기로는 아직 문헌에 언급된 바가 없다.

본 연구에 사용된 정수최적해 도출을 위한 실험적인 Branch-and-Price 알고리즘은 테스트 대상인 8개의 Asymmetric 문제 중 수요지 50곳 이하인 문제에서만 최적화에 성공하였다. 하한값을 강화하는 Branch and Cut and Price 알고리즘과 효과적인 branching 기법의 개발은 향후 주요 연구과제이다.

부 록 A: 열생성 하위문제

$n(k)$ 를 차량 k 의 경로에 포함된 수요지의 수라고 하고, $U(k)$ 를 $n(k)$ 의 상한, $L(k)$ 를 $n(k)$ 의 하한이라고 하자. 즉, $L(k) \leq n(k) \leq U(k)$ 이 성립한다. $n(k,s)$ 는 차량 k 의 s 번째 분할경로에 포함된 수요지의 수이고, $U(k,s)$ 는 $n(k,s)$ 의 상한, $L(k,s)$ 는 $n(k,s)$ 의 하한이라고 하자. 즉, $L(k,s) \leq n(k,s) \leq U(k,s)$ 이 성립된다.

가. $L(k)$ 와 $U(k)$

다음은 차량 $k^o \in \{1, \dots, K\}$ 의 $U(k^o)$ 를 구하기 위한 수리계획모형이다.

$$(GA) \quad U(k^o) = \text{maximize} \quad \sum_i Y_{k^o i}$$

subject to

$$\sum_i q_i Y_{ki} \leq Q_k \quad \forall k$$

$$\sum_k Y_{ki} = 1 \quad \forall i$$

$$Y_{ki} \in \{0,1\} \quad \forall k \forall i.$$

변수 Y_{ki} 는 수요지 i 가 차량 k 의 경로에 포함되면 1, 아니면 0의 값을 갖는다. (GA)는 Generalized Assignment 문제이며, $L(k)$ 는 목적함수의 최소화로 구한다.

나. $s(k)$ 와 $L(k, s)$ 와 $U(k, s)$ 구하는 모듈

1. if $U(k) \leq B$, then $s(k) = 1$.

// $U(k)$ 가 B 보다 크지 않을 때는 차량 k 의 경로는 분할되지 않음

2. if $U(k) > B$, then $s(k) = \left\lceil \frac{U(k) - B}{B - 1} \right\rceil + 1$

3. for $s = 1$ to $s(k) - 1$, $U(k, s) = B$

for $s = s(k)$, $U(k, s) = U(k) - B - (B - 1) \times (s - 2) + 1$

4. if $L(k) \leq B$, then

for $s = 1$, $L(k, s) = L(k)$

for $s = 2$ to $s(k)$, $L(k, s) = 1$

5. if $U(k) > B$, then $tmpr = \left\lceil \frac{L(k) - B}{B - 1} \right\rceil + 1$

for $s = 1$ to $tmpr - 1$, then $L(k, s) = B$

for $s = tmpr$, then $L(k, s) = L(k) - B - (B - 1) \times (s - 2) + 1$

for $s = tmpr + 1$, then $L(k, s) = 1$

다. Procedure GenColumn {

// $\theta^t(k, s, j)$ 는 차량 k 의 s 번째 분할경로에서 수요지 j 에서 시작해서 t 개의 수요지를 포함하는

// tail 부분의 reduced cost의 하한값이다.

// Initialization

for $k = 1$ to K

for $s = 0$ to $s(k)$


```

    for  $j = 1$  to  $N$ 
      for  $t = 2$  to  $B - 2$ 
         $\theta^t(k, s, j) = \infty$ ;

    for  $k = 1$  to  $K$ 
      GetBound ( $k$ );

// generating  $\theta^o(k, s, j)$ 
    for  $k = 1$  to  $K$ 
      for  $s = 0$  to  $s(k)$ 
        for  $j = 1$  to  $N$  {
           $\theta^o(k, s, j) = 0$ ;
          for  $t = 2$  to  $B - 2$ 
            if  $\theta^t(k, s, j) < \theta^o(k, s, j)$ , then  $\theta^o(k, s, j) = \theta^t(k, s, j)$ ;
        }

    for  $k = 1$  to  $K$ 
      for  $s = 0$  to  $s(k)$ 
        SubProblem ( $k, s$ );

//end GenColumn

Procedure GetBound ( $k$ ) {
  for  $j = 1$  to  $N$ 
    Bound ( $k, 1, \{j\}, j, 0, j$ );
}

Procedure Bound ( $k, t, W, w, C, j$ ) {
  for  $i \in \{1, \dots, N\} - W$  {
     $tmpc = C + d_{wi} + d_{io} - q_i \gamma_k - \beta_i$ ;
    if  $tmpc < \theta^{t+1}(k, 0, j)$ , then  $\theta^{t+1}(k, 0, j) = \theta^{t+1}(k, s(k), j) = tmpc$ ;
    for  $s = 1$  to  $s(k) - 1$  {
       $tmp(s) = C + d_{wi} + \pi_{isk}$ ;
      if  $tmp(s) < \theta^{t+1}(k, s, j)$ , then  $\theta^{t+1}(k, s, j) = tmp(s)$ ;
    }
  }
}

```

```

    }
    if  $t \leq B-4$ , then {
         $newC = C + d_{wi} - q_i \gamma_k - \beta_i$ ;
        Bound ( $k, t+1, W \cup i, i, newC, j$ );
    }
}

for  $s = 0$  to 1
    Algorithm SubProblem ( $k, s$ ) {
         $minDual = 0$ ;
         $C = -\alpha_k$ ;
        MinCoat ( $k, s, 0, \emptyset, 0, C$ );
    } // end SubProblem

for  $s = 2$  to  $s(k)$ 
    Algorithm SubProblem ( $k, s$ ) {
         $minDual = 0$ ;
        for  $j = 1$  to  $N$  {
             $C = -\beta_j - q_j \gamma_k - \alpha_k - \pi_{jsk}$ ;
            MinCost ( $k, s, 1, \{j\}, j, C$ );
        }
    } //end SubProblem

Procedure MinCost ( $k, s, t, W, w, C$ ) {
    for  $i \in \{1, \dots, N\} - W$ 
    {
        if  $t+1 \geq L(k, s)$ , then Update ( $k, s, i, C$ );
        if  $t+2 \leq U(k, s)$ , then
        {
             $newC = C + d_{wi} - q_i \gamma_k - \beta_i$ ;
            if  $U(k, s) - t > B-2$  or  $newC + \theta^0(k, s, j) < minDual$ , then
                MinCost ( $k, s, t+1, W \cup i, i, newC$ );
        }
    }
}

```

```

    }
} //end MinCost

Procedure Update (k, s, w, C) {
    if s = 0 or s(k), then tmp = C + dui + dio - βi - qiγk;
    else tmp = C + dui + πisk;
    (here update the sorted list for multiple pricing);
    if tmp < minDual, then minDual = tmp;
} //end Update

```

부 록 B: Route-Splitting 모형의 하한값

$z(\cdot)$ 을 모형 (·)의 최적값이라고 하면 $z(\text{LRS}(R))$ 은 $z(\text{LRS})$ 의 상한이다. 다음의 모형에서 LRS의 쌍대변수 β_i , γ_k , α_k 와 π_{isk} 의 벡터를 u 라고 하면,

$$\begin{aligned}
 (\text{dRS}(u)) \quad \min \quad & \sum_{r \in R} d_r x_r + \sum_i \beta_i (1 - \sum_{r: i \in B(r)} x_r) + \sum_k \gamma_k (Q_k - \sum_{r \in R(k)} q_r x_r) \\
 & + \sum_k \alpha_k (1 - \sum_{r \in R(k,0)} x_r - \sum_{r \in R(k,1)} x_r) \\
 & + \sum_k \sum_{s=2}^{n(k)} \sum_i \pi_{isk} \left(\sum_{r \in R(k,s-1)|e(r)=i} x_r - \sum_{r \in R(k,s)|b(r)=i} x_r \right) \quad (8)
 \end{aligned}$$

subject to

제약조건식 (3) — (6),

그렇다면 $z(\text{dRS}(u))$ 는 $\gamma_k \leq 0$ 인 어떤 u 에 대해서도 $z(\text{RS})$ 의 하한값이 된다. 다음의 $\text{DRS}(u)$ 는 $\text{dRS}(u)$ 와 같은 목적함수를 가진다.

$$(\text{DRS}(u)) \quad \min \quad (8)$$

subject to

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,1)} x_r = 1 \quad \forall k$$

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,2)} x_r = 1 \quad \forall k$$

...

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,s(k))} x_r = 1 \quad \forall k$$

$$x_r \in \{0,1\} \quad \forall r \in R.$$

그렇다면 $z(\text{DRS}(u))$ 는 $z(\text{dRS}(u))$ 의 하한값이며, 따라서 $z(\text{RS})$ 의 하한값이 된다. 우리는 라그랑지 상수인 u 의 값으로 LRS 의 쌍대변수값을 사용한다. 다음의 관계식은 잘 알려져 있다.

$$z(\text{RS}) \geq z(\text{DRS}) \equiv \max_u z(\text{DRS}(u)) \geq z(\text{LRS}).$$

dRS(u)의 목적함수에서 $\sum \beta_i + \sum \gamma_k Q_k + \sum \alpha_k$ 는 u 가 주어진 상황에서 상수이며 쌍대정리에 의해서 $z(\text{LRS}(R))$ 과 동일하다. 상수부분을 제외한 부분은 각 차량에 하나씩 K 개의 문제로 분리된다. 차량 k 에 해당되는 부분은:

$$z_k = \min \sum_{r \in R} d_r x_r - \sum_i \beta_i \sum_{r|i \in B(r)} x_r - \gamma_k \sum_{r \in R(k)} q_r x_r - \alpha_k \left(\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,1)} x_r \right)$$

$$+ \sum_{s=2}^{n(k)} \sum_i \pi_{isk} \left(\sum_{r \in R(k,s-1)|e(r)=i} x_r - \sum_{r \in R(k,s)|b(r)=i} x_r \right)$$

subject to

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,1)} x_r = 1$$

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,2)} x_r = 1$$

...

$$\sum_{r \in R(k,0)} x_r + \sum_{r \in R(k,s(k))} x_r = 1$$

$$x_r \in \{0,1\} \quad \forall r \in R(k)$$

위의 차량 k 에 대한 모형은 또한 $s = 0, \dots, s(k)$ 에 대한 모형으로 분리된다. $s = 0$ 과 $s = 1$ 의 경우는

$$z_{sk} = \min \sum_{r \in R} d_r x_r - \sum_i \beta_i \sum_{r|i \in B(r)} x_r - \gamma_k \sum_{r \in R(k)} q_r x_r - \alpha_k \sum_{r \in R(k,s)} x_r$$

subject to

$$\sum_{r \in R(k,s)} x_r = 1$$

이며, $s = 2, \dots, s(k)$ 의 경우는 각각

$$z_{sk} = \min \sum_{r \in R} d_r x_r - \sum_i \beta_i \sum_{r|i \in B(r)} x_r - \gamma_k \sum_{r \in R(k)} q_r x_r$$

$$+ \sum_i \pi_{isk} \left(\sum_{r \in R(k, s-1)|e(r)=i} x_r - \sum_{r \in R(k, s)|b(r)=i} x_r \right)$$

subject to

$$\sum_{r \in R(k, s)} x_r = 1$$

이다. 여기서 각 z_{sk} 는 차량 k 의 s 번째 분할경로의 변수의 최소 reduced cost이며, 우리의 열생성 과정의 하위문제에서 얻어지는 것이다. 정리하면,

$$z(\text{DRS}(u)) = \sum_k \min(z_{0k}, \sum_{s=1}^{s(k)} z_{sk}) + z(\text{LRS}(R))$$

로서 식의 요소들은 현 iteration의 주문제와 하위문제에서 얻는 값들이다. 한편 $\text{DRS}(u)$ 의 integrality property에 의해서 $z(\text{DRS}) = z(\text{LRS})$ 가 성립한다. 하한값 $z(\text{DRS}(u))$ 는 열생성 프로세스에서 단조적으로(monotonously) 증가하지는 않지만, $z(\text{LRS})$ 에 수렴하며, 열생성 프로세스의 종료시점을 결정하는데 이용된다.

참고 문헌

- Agarwal, Y., K. Mathur, and H. M. Salkin (1989), "A Set-Partitioning-Based Exact Algorithm for the Vehicle Routing Problem," *Networks*, 19, 731-749.
- Bodin, L., B. Golden, A. Assad, and M. Ball (1983), "Routing and Scheduling of Vehicles and Crews: The State of the Art," *Comput. Opns Res.* 10, 63-211.
- Christoffides, N., A. Mingozzi, and P. Toth (1981), "Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations," *Mathematical Programming*, 20, 255-282.
- CPLEX Optimization, Inc (1992), *Using the CPLEXTM Callable Library and CPLEXTM Mixed Integer Library*. Incline Village, NV.
- Dell' Amico M. and P. Toth (2000), "Algorithms and codes for dense assignment problems: The state of the art," *Discrete Applied Mathematics*, 100, 17-48.
- Desrochers, M., J. Desrosiers, and M. Solomon (1992), "A New Optimization Algorithm for the Vehicle Routing Problem with Time Windows," *Operations Research*, 40, 342-354.
- Desrochers, M., and F. Soumis (1989), "A Column Generation Approach to the Urban Transit Scheduling Problem," *Transportation Science*, 23, 1-13.
- Dongarra, J. J. (1988), "Performance of various computers using standard linear

- equations software," *Technical Report CS-89-85*, University of Tennessee, Knoxville.
- Eilon, S., C. Watson-Gandy, and N. Christofides (1971), *Distribution Management: Mathematical Modeling and Practical Analysis*. Pages 196—201. Hafner, New York.
- Fischetti, M., P. Toth, and D. Vigo (1994), "A Branch-and-Bound Algorithm for the Capacitated Vehicle Routing Problem on Direct Graphs," *Operations Research*, 42, 5, 846—859.
- Fisher, M. L. (1994), "Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees," *Operations Research*, 42, 626—642.
- Held, M., and R. M. Karp (1970), "The Traveling-Salesman Problem and Minimum Spanning Trees," *Operations Research*, 18, 1138—1162.
- Held, M., and R. M. Karp (1971), "The Traveling-Salesman Problem and Minimum Spanning Trees: Part II," *Mathematical Programming*, 1, 6—25.
- Laporte, G., Y. Nobert, and M. Desrochers (1985), "Optimal Routing under Capacity and Distance Restrictions," *Operations Research*, 33, 1050—1073.
- Laporte, G., H. Mercure and Y. Nobert (1986), "An Exact Algorithm for the Asymmetrical Capacitated Vehicle Routing Problem," *Networks*, 16, 33—46.
- Laporte, G., and Y. Nobert (1987), "Exact Algorithms for the Vehicle Routing Problem," *Annals of Discrete Mathematics*, 31, 147—184.
- Laporte, G. (1992), "The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms," *European Journal of Operations Research*, 59, 345—358.
- Lasdon, L. (1971), *Optimization Theory for Large Systems*, Macmillan Company, New York, NY, p164.
- Lavoie, S., M. Minoux, and E. Odier (1988), "A New Approach for Crew Pairing Problems by Column Generation with an Application to Air Transportation," *European Journal of Operations Research*, 35, 45—58.
- Miller, D. L. (1995), "A matching based exact algorithm for capacitated vehicle routing problems," *ORSA Journal on Computing*, 7, 1—9.
- Nemhauser, G. L. (1994), "The Age of Optimization: Solving Large-Scale Real World Problems," *Operations Research*, 42, 5—13.
- Toth, P. and D. Vigo (2002), "Models, relaxations and exact approaches for the capacitated vehicle routing problem," *Discrete Applied Mathematics*, 123, 487—512.
- Vigo, D. (2000), *VRPLIB: A Vehicle Routing Problem Library*.
http://www.or.deis.unibo.it/research_pages/ORinstances/VRPLIB/VRPLIB.html.