韓國經營科學會誌
第19卷 第1號
1994年 4月

차량경로문제(VRP)의 최적루트 설계를 위한 알고리듬 개발에 관한 연구

이규헌*

A Study of the Development of Algorithm for Optimal Route Design of the Vehicle Routing Problems

Kyu – Heon Lee*

Abstract

This paper is concerned with the development of tree-search algorithm for the exact solution to the vehicle routing problem (VRP), where set of vehicles of known capacity based at depot, have to be routed in order to supply customers with known requirements. What is required is to design routes, so that the total cost(i. e. total route length or time duration, ect.) is minimized.

For obtaining the exact solution, the most important factors are the value of bound and branching strategy. Using the bound based on with bound ascent procedures from subgradient and state-space ascents, the incorporation of bounds into tree search algorithm to solve the problem is shown.

Computational results of the corresponding algorithm show that VRPs with up to 40 customers can be solved optimally with this algorithm.

* 육군사관학교 경제 · 경영학과 부교수

1. 서 론

지리적으로 산재해 있는 고객(customer) 또는 지점에서 요구하는 수요량을 공급할 때, 분배센터(depot)에 있는 차량들에 의해 최소비용으로 분배해야 하는 문제가 발생한다. 이때 비용을 최소화하기 위해서는 최적의 분배루트(차량경로)가 설계되어야 하며, 반드시 고객의 수요충족이라는 분배목적을 달성해야 한다. 여기서 차량은 반드시 분배센터를 출발하여 다시 분배센터로 돌아와야 한다는 가정을 한다. 이러한 차량경로문제 (VRP : Vehicle Routing Problem)는 차량에 의해 고객을 방문하게 되는 모든 문제를 포함하는 일반적인 명칭으로 사용되며, 차량일정계획[9,11,14], 차량배치계획[4,10,16,22] 또는 배송문제[2,17,24]등으로 표현되고 있다. 그러나 차량경로 문제는 물적 분배와 관련된 것 뿐 아니라 현실적 상황에서 흔히 나타날 수 있는 여러가지 문제들, 즉 우편물수집, 공중전화박스의 동전회수, 스쿨버스의 학생통학, 의사의 왕진경로, 설비유지 및 보수를 위한 순회등 일상생활이나 현실적 상황에서 수집 또는 분배 문제가 발생하는 한 다양한 분야에서 발생한다. 무수히 많이 발생될 수 있는 현실적 상황의 관점에서 보면, 차량경로문제의 해결을 위한 목표 및 제약조건이 방대할 수 밖에 없으며, 이때문에 문제의 모형화 및 해결이 쉽지 않다. 그러므로 아직도 현실적인 문제를 해결할 수 있는 일반화모형이 개발되지 못하고 있는 것이다.

차량경로문제는 여행자문제(TSP : Travelling Salesman Problem)를 일반화시킨 것이다. 즉, 여행자문제는 차량경로문제에서 단 한대의 차량만이 존재하는 경우를 생각할 수 있다. 이러한 여행자문제는 수백개의 고객이 존재하는 현실적 문제를 해결 할 수 있는 일반적인 알고리듬이 개발되어왔으나[5,25], 차량경로문제의 경우 여행자문제의 경우와 같이 최적해를 구하는 일반적인 알고리듬이 아직 존재하지 않는다. 즉, 문헌상에 나타난 결과를 보면, 특별한 조건을 추가한 경우를 제외하면[21], 겨우 20여개의 고객을 갖는 규모의 문제에 대해 최적해를 구하는 일반적 알고리듬이 개발되었을 뿐이다[6,16]. 그러므로 차량경로문제는 주로 발견적 해법(heuristic approach)이 사용되고 있으며, 이 방법으로는 수백개의 고객을 갖는 규모의 문제를 해결하고 있다. 따라서 차량경로문제의 최적해를 구하는 일반적인 알고리듬을 개발 하여 보다 현실적인 문제를 해결할 수 있도록 하는 노력이 필요하다고 생각된다.

그러므로 본 연구는 차량경로문제 해결의 가장 중요한 두 요인인 한계값과 분단탐색법 중 분단탐색법을 개선한 알고리듬을 개발하여 보다 개선된 최적해의 해결방법을 제시하고자 한다.

2. 문제정의 및 한계값

기본차량경로문제는 다음과 같이 정의된다. 지점의 집합 X와 아크의 집합 A로 정의된 그래프 $G=(X,A)$ 를 생각해 보자. $X=\{x_i \mid i=1,\cdots,n\}$이 n개의 지점(분배센터와 수요지점 : 고객)으로 이루어진 집합이라 하면, 고객은 $i=2,\cdots,n$ 으로 표현되고 $i=1$ 은 분배센터(depot)가 된다. 또 하나의 집합 $V=\{v_k \mid k=1,\cdots,m\}$은 분배센터에 있는 이용가능한 차량의 집합이고, $k=1,\cdots,m$ 은 차량의 수이다.

수요지점인 고객 i는 요구량(물적분배량) q_i를 갖는다. 이때 i와 j사이의 분배비용은 c_{ij} 로 표시되며, 이는 고객사이의 거리 또는 시간이 된다. 차량의 적재량을 Q_k 라 하면, 고객들과 차량은 각각 q_i 와 Q_k 의 감소순으로 배열될 수 있다.

여기서 기본 차량경로문제는 모든 고객들의 수요를 충족시키면서 수송거리를 최소화할 수 있도록 차량의 분배경로(차량 1대는 1개 경로를 담당하며, 분배센터에서 출발하여 다시 되돌아온다)를 설계하는 것이다. 기본 차량경로문제는 현실적으로 나타날 수 있는 다양한 형태의 문제를 규정하기위한 제약조건들과 확장된 모형은 고려하지 않는다. 현실적인 문제에 접근하기 위해 시도된 제약조건 추가 및 모형의 확장은 Christofides등의 연구결과에 다양하게 나타나 있다[19].

위에서 정의된 그래프 G에서 각 아크에 관련된 비용 (거리 또는 시간등)을 $[c_{ij}]$ 로 표시하고, c_{ij}와 $c(x_i,x_j)$를 같은 의미로 상호 교환가능하다고 하면, 그래프 G에 관한 전형적인 경로문제는 G의 모든 지점을 통과하면서 최소비용의 경로설계가 요구되는 여행자문제(TSP)라 할 수 있다.

경로문제를 해결하는데 있어서 가장 성공적인 방법중의 하나는 분단탐색법(branch and bound method) 이며, 여기서 완성된 알고리듬의 효율성을 결정해주는 가장 중요한 인자는 한계값이다. 한계값을 구하는 일반적인 방법은 Lagrangean relaxation 방법이다[15]. 이 방법은 사용가능한 한계값을 구하기 위한 방법중의 하나에 불과하지만, 차량경로문제 등 여러 종류의 조합최적화문제(combinatorial optimization problem)에 잘 적용되고 있다[12]. 그러나 기본적인 경로문제에 제약조건, 예를들어 여행자문제에서의 시간이라든지, 차량경로문제의 차량적재량 및 수요량 변화 등을 추가하고자 하면, 최초모형이 지닌 구조를 파괴하는 경향이 있다.

이러한 어려움때문에 경로문제(TSP, VRP 등)를 해결하는데 있어서 새로운 대체방법이 Christofides등에 의해 개발되었다[7]. 이 방법은 다음의 두 가지 사실에 근거를 두고 있다. 첫째, 모든 경로문제는 근본적으로 추가된 제약조건을 가진 그래프상의 최단경로문제(shortest path problem)의 하나이다. 둘째, 제약조건이 추가되었거나 확장된 모형인 경우에 최단경로문제의 일반적인 해결절차는 동적계획법이라는 것이다.

앞서 언급했듯이 여행자문제는 경로문제의 하나이고, 그래프 G의 모든 지점을 단 한번 통과하는 하나의 최단경로를 찾는 문제이며, 또한 차량경로문제는 여행자문제를 일반화시킨 조합최적화 문제의 하나다. 이러한 조합최적화 문제는 state-space 그래프상에 지점의 수가 무수히 많기 때문에 동적계획법 자체만으로는 해결될 수 없다[3]. 그러므로 일반적인 완화방법 즉, 상태공간완화법(state-space relaxation)이 제안되었으며 이에 의해 동적계획법의 순환등식과 관련된 상태공간이 완화 즉 상태의 수가 제거된다[7]. 상태공간완화는 정수계획법에서의 라그랑지안 완화를 유추한 것이다. 정수계획모형에서의 제약조건은 동적계획법에서는 상태변수로 나타난다. 그러므로 제약조건의 완화는 상태공간의 완화에 상응한다. 이때, 완화된 순환등식의 해는 실제 최적해에 대해 한계값(최소화문제의 경우 하한값, 최대화문제의 경우 상한값)을 갖게 된다[8]. 이러한 완화된 순환등식의 해인 한계값은 경로문제의 해를 구하기 위한 분단탐색법을 이용한 알고리듬의

분단의 한계를 제공하게 된다. 여기서 차량경로문제의 대표적인 한계값을 구하는 방법[6,20]을 이용하여 즉, 차량경로문제의 동적계획모형으로부터 직접 구한 하한값을 LB_1, 동적계획모형과 정수계획모형을 이용한 하한값을 LB_2 라고 한다면, LB_1과 LB_2의 값은 벌칙부과방법(penalty procedures)과 부분기울기법(subgradient method)를 이용하여 개선된다[18,23]. 이와같이 개선된 최종한계값 즉, LB_1(직접한계값)과 LB_2(간접한계값)은 본 논문에서 개발하려는 나무탐색(tree-search)알고리듬을 적용하여 차량경로문제의 최적해를 구하는데 사용된다.

3. 최적경로 설계를 위한 tree-search 알고리듬

차량경로문제의 해를 구하는데 있어서 분단탐색법을 이용하여 모든 실현가능한 해들을 검토하는 것은 끝없는 작업이 될 수도 있으므로 실현가능한 범위내로 계산에 소요되는 노력을 줄여야 한다. 이때 나무탐색 알고리듬의 효율성은 앞에서 언급한 한계값의 질과 분단전략(branching strategy)에 의해 결정된다. 이러한 두 요인중 분단탐색법을 이용하여 최적해를 구하기 위한 차량경로문제의 나무탐색 알고리듬을 개발한다. 단, 한계값은 이미 개발된 알고리듬[6,20]에 의해 얻어진 값을 이용한다.

3.1 기본 나무탐색 알고리듬

어떤 나무탐색 알고리듬이든 그 기본은 첫째, 모든 가능한 루트들의 집합을 보다 작은 부분집합으로 나누고, 다음에 가장 나은 루트의 비용에 관한 하한값을 각 부분집합에 대해 계산하는 것이다. 이러한 하한값을 계산하는 목적은 첫째로 루트의 부분집합을 분할하는 지표역할을 하는데 사용되도록 하는 것이고, 둘째로 수많은 탐색나무를 필요한 정도로 한계를 설정하면서 최적 루트를 찾아내는 것이다. 이러한 탐색나무를 구성하는데 있어서 분단전략(branching strategy)을 고려하는 것이 중요하다. [그림 1]은 다음 항에서 설명되는 분단전략을 이용하여 차량경로문제에 대한 기본적인 나무탐색(basic depth-first tree search) 알고리듬의 흐름도를 보여주고 있다. [그림 1]에서의 다양한 항(boxes)의 기능은 아래와 같이 보다 상세히 설명될 수 있다.

[step 1] 초기화 : 비용행렬 C를 읽어들이고, 분단이 계속되는데 따르는 분단지점에 대한 숫자부여(N) 및 나무의 깊이의 수준(level of tree)에 대한 숫자부여(L)를 한다. 그리고 최적해(Z_{opt} : 지금까지의 가장 최선의 루트에 대한 비용)를 Z_{UB}(탐색적 상한값 : heuristic upper bound)로 설정하고, 분단숫자상에서의 해(Z(N))를 최초노드(N=0)에서의 하한값인 LB_N으로 설정한다.

[step 2] 분단실현시험(branching test) : 다음 분단을 위해 선택될 고객(지점) x_i (i=2, , n)의 집합이 있는지 또는 없는지를 검토한다. 만일 어떤 루트에 포함되지 않은 고객집합이 비어있다면, GOTO [step 10], 그렇지 않으면 계속한다.

[step 3] 분단(branching) : 분단전략에 따른 규칙에 의거하여 다음 단계로 분단을 하기 위

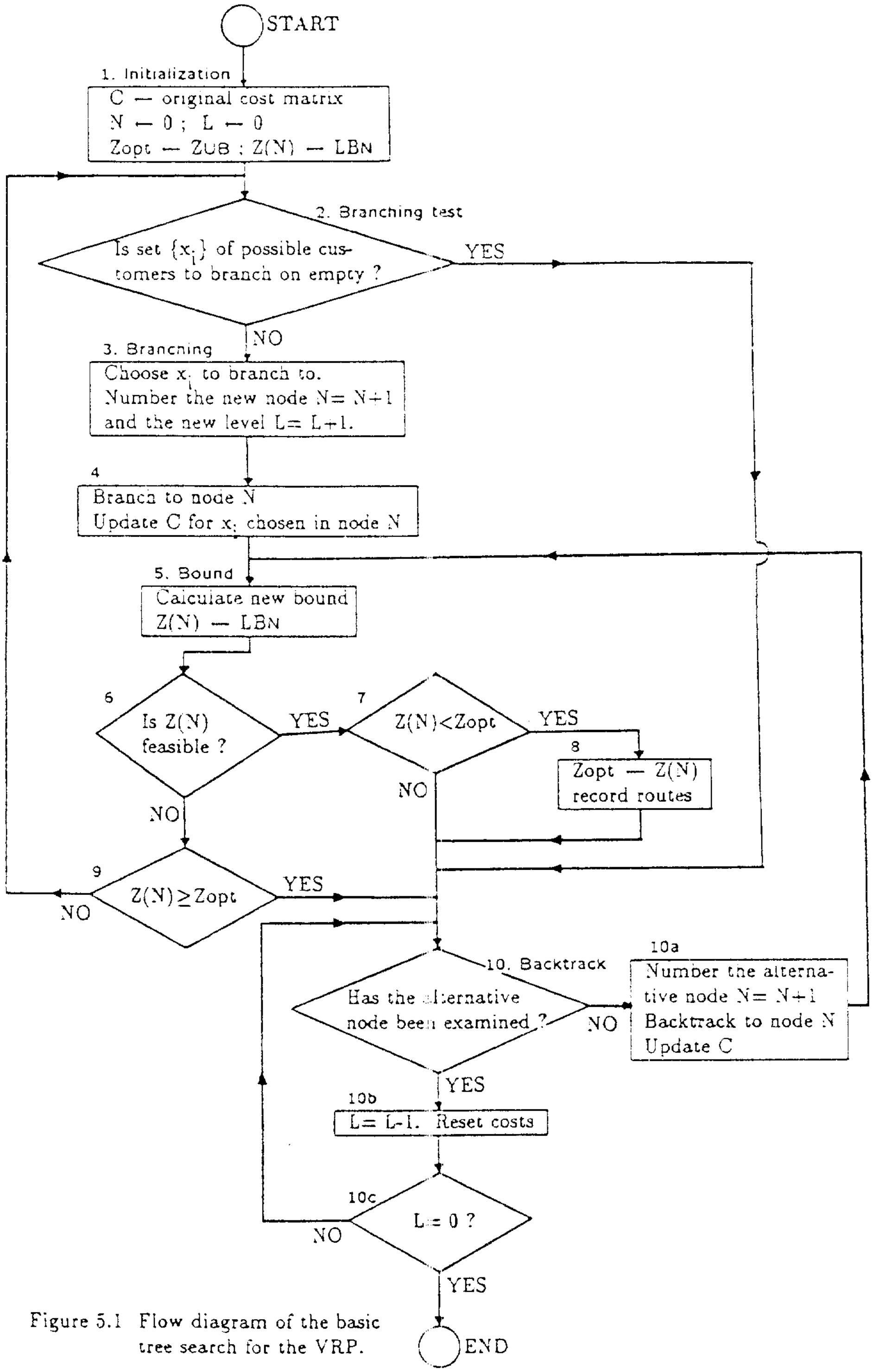


Figure 5.1 Flow diagram of the basic tree search for the VRP.

[그림 1] 차량경로문제에 대한 기본 나무탐색 알고리듬의 흐름도

한 하나의 고객을 선택하고, 분단지점의 수(N)와 나무깊이 수준(L)에 대한 숫자를 부여한다.

[step 4] 분단지점 N에서의 전진분단(forward branching)을 하고 비용행렬(C)의 요소들을 수정한다.

[step 5] 한계값 : 분단지점(N)에서의 하한값을 계산하고 $Z(N) = LB_N$ 으로 기록한다.

[step 6,7,8,9] 만일에 [step 5]에서 구한 한계값에 상응하는 해가 실현가능하고 그때의 비용이 Z_{opt} 보다 작다면, 그 해의 값을 Z_{opt} 로 기록하고 다음 단계의 분단을 위해 GOTO [step 10]. 만일 비용이 Z_{opt} 보다 크거나 같으면 바로 GOTO [step 10]. 또한 만일에 비용이 Z_{opt} 보다 작고 실현가능해가 아니면 전진분단을 계속하기 위해 GOTO [step 2].

[step 10] 후진분단을 위한 역추적(Backtrack) ; [step 10a, 10b, 10c] : 만일 현재의 분단지점(N)에 대한 대안(alternative : 선택되지 않은 고객에 대한 분단)이 검토되지 않았나면, 그 분단지점의 해에 대한 평가를 한다. 여기서 N=N+1로 하고, 이 지점에서의 비용행렬을 수정한 다음 GOTO [step 5]. 만일 현재 분단지점에 대한 대안을 검토했다면, L=L−1로 하고, L=0이면 알고리즘의 작업은 끝내고 ; 그렇지 않으면 바로 전단계의 비용(현단계에서 사용한 비용행렬을 이용하여 구한 값)을 다시 기록하고 [step 10]을 반복한다.

3.2 분단전략

분단전략(다음 단계의 분단을 검토 분석하기 위해 어떤 고객을 선택할 것인가를 결정하는 문제)은 하나의 부분적으로 완성된 루트($x_j, x_k, \cdots, x_i$)를 확장하기 위하여 탐색나무의 어느 분단지점에서 전진분단을 하기 위해 선택되는 아크 (x_i, x_j)를 기초로 한다. 또 다른 대안(alternative)으로써의 분단전략은 현 분단지점에서 전진분단을 위해 선택되었던 아크(x_i, x_j)를 부분적으로 완성된 루트의 확장가능성이 있는 것으로 간주하지 않는 경우이다.

분단을 위해 이러한 아크(x_i, x_j)를 선택하는데 있어서 (이것은 고객 x_j 가 x_i 바로 뒤에 위치하도록 루트를 확장하는데 이용되는 것을 의미함) 다음과 같은 분단규칙(branching rule)이 적용된다.

3.2.1 첫 루트의 시작

어떤 루트를 구성하기 시작하는 경우에, 전진분단을 위하여 선택된 아크(x_i, x_j)는 아직 선택되지 않은 고객(x_j)중에서 분배센터로부터 가장 가까운 아크가 될 것이다. (여기서 만일 한개이상의 아크가 나타날 경우에는 수요량이 가장 큰 고객을 선택한다). 이러한 고객은 루트를 구성하는데 있어 분배센터에서 출발하여 루트 첫번째 요소가 된다.

3.2.2 부분적으로 완성된 루트의 확장

만일 어떤 부분적으로 완성된 루트가 반드시 확장되어야 한다면, 이때 전진분단을 위해 선택되어야 할 고객은 차량의 적재능력을 위배하지 않으면서 루트의 마지막 고객으로부터 가장 가까운 곳에 위치한 지점(고객)이 된다. 분단지점으로서 이러한 유형의 고객을 선택함으로

써, 분단대안(선택된 고객을 갖는 부분적으로 완성된 루트의 확장을 취소하는 것)은 보다 높은 하한값을 갖게 되며, 현재의 분단지점으로부터 바로 전 분단지점으로 되돌아가게 된다.

3.2.3 부분적으로 완성된 루트의 끝

어떤 부분적으로 완성된 루트는 분단의 결과 어떤 아크가 (x_j, x_1)이 되는 경우에 끝나고, 다시 새로운 루트를 시작하게된다.

3.2.4 새로운 루트의 시작

현재까지의 분단이 하나의 완전한 루트들의 집합을 형성했을 경우에 다시 새로운 루트가 시작되어야만 한다. 이 경우에, 분단을 위해 선택된 고객은 아직까지 어떤 루트에도 포함되지 않은 고객중의 하나가 된다. 이때 그 고객은 차량적재능력을 위반하지 않으면서 분배센터로부터 가장 가까운 지점에 있는 것이 된다.

만일 어떤 한 아크(x_i, x_j)가 부분적으로 완성된 루트$(x_1, x_k, \cdots, x_i)$를 확장하기 위하여 나무탐색의 한 분단지점에서 선택되었다면, 그 비용은 다음과 같이 변화된다.

$$c_{jl} = c_{lj} = \quad (l=2,\cdots,n \text{ 과 } l \neq i,j \text{ 일때}),$$

또한 부분적으로 완성된 루트의 확장이 가능하지 않다고 생각된 경우, 즉 아크(x_i, x_j)가 제외된 경우인 분단대안에 대하여 그 비용은 다음과 같이 변화된다.

$$c_{ij} = c_{ji} = \infty$$

3.2.5 분단의 맺음

탐색나무의 어느 한 분단지점은 다음과 같은 상황이 발생할 때 끝을 맺게 되어 후진분단의 시작 또는 알고리듬의 탐색과정을 마치게 되는데, 다음과 같은 세가지 경우에 한한다.

(i) 어느 한 단계에서 하한값이 현재의 상한값(지금까지 최선의 해)보다 크거나 또는 같을 경우에 후진분단이 발생하게된다.

(ii) 만일 어떤 부분문제의 하한값에 상응하는 해가 실현가능한 해가 될 경우이다.

(iii) 만일 탐색나무의 최저점에 도달하였거나 더 이상 검토해야 할 지점(고객의 방문지점)이 존재하지 않는 경우이다.

3.3 알고리듬에 대한 분석

앞에서 설명한 알고리듬의 효율성을 제고시키기 위하여 다음과 같은 몇가지 사항을 고려해 볼 수 있다.

① 하나의 루트가 완성될 때마다, 최초의 문제로부터 완성된 루트에 포함된 고객들을 제외시킴으로써 문제의 규모를 줄일 수 있다. 이 경우에 문제의 하한값을 계산할 때 감소된 규모의 문제에 대하여 산출하고 완성된 루트에 해당하는 비용부분을 합산하면 된다.

② 여기서 개발된 알고리듬에 의해 도출된 초기 하한값의 실험결과는 최적해에 상당히 근접한 값(통상 4%이내)을 갖으므로, 초기 상한값을 $k*ZL$ (예, $k=1.04$) 의 값으로 대체하여 사용할 수 있다. 이 경우에 초기 상한값이 탐색적 방법(heuristic method)에 의해 구한 상한값보다 상당히 최적해에 근접하기 때문에,

나무탐색법에서 분단지점의 수를 크게 줄일 수 있어 알고리듬의 효율성을 높여준다.

③ 마지막 루트를 형성하기 위한 절차를 피하기 위해서 여행자경로문제(TSP)의 알고리듬을 적용할 수 있다. 또한 하나의 루트가 완성될 때마다, 이제까지의 부분해가 최적해인지 아닌지를 결정하기 위해서 역시 여행자 경로문제 알고리듬을 적용할 수 있다.

④마지막으로, 어떤 분단지점에서 하한값과 상한값 사이에 갭(gap)을 고려할 수 있다. 예를 들어, 만일 최초의 비용행렬 요소값이 정수라면, 그 실현가능해 역시 정수이어야만 한다. 그러므로 만일 어떤 분단지점에서의 하한값이 상한값과 상한값 −1 (즉, $0 < Z_{opt} - ZL < 1$) 사이에 있다면, 더이상 최적해가 발견될 수 없으므로 바로 이 지점으로부터 후진분단탐색을 할 수 있게 된다.

이상의 네가지 요소를 고려함으로써 차량경로문제의 나무탐색 알고리듬의 효율을 상당히 높여줄 수 있게 된다. 앞의 알고리듬과 위의 네가지 요소를 종합하여 최종적으로 알고리듬이 완성된다. 여기서 알고리듬 1은 앞의 분단전략과 직접 하한값(LB1) 을 결합하여 형성된 나무탐색 알고리듬이며, 알고리듬 2는 간접하한값(LB2)과 결합하여 알고리듬 1과 같은 방법으로 형성된 것이다. 다음 항에서 이와 같은 두개의 알고리듬을 이용하여 간단한 예를 들어 보고자 한다.

4. 알고리듬 적용 예

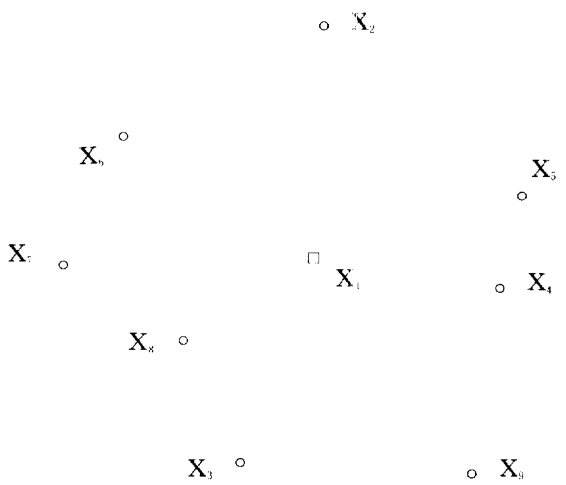
예제로써 하나의 분배센터, 8개의 분배지점(고객)과 3대의 차량, 그리고 비용(여기서는 거리를 의미)행렬을 갖는 차량경로 문제를 고려해 보자. 이 예제에 대한 비용표와 그래프가 〈표 1〉과 [그림 2]에 주어져 있다. 여기서 x_1은 분배센터를 의미하고 $x_2, \cdots, x_9$ 은 고객을 나타낸다. 차량의 적재능력은 Q=6 단위로 하며, 고객의 수요량은 아래 표와 같이 주어진다고 가정한다. 또한 총수송량 Q=15 가 된다.

x_i =	1	2	3	4	5	6	7	8	9
q_i =	0	2	3	1	1	2	1	3	2

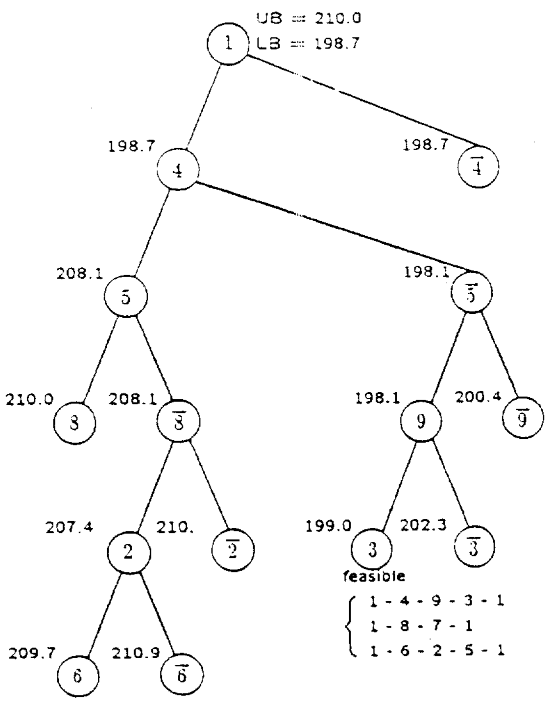
〈표 1〉 비용(거리) 행렬 [C_{ij}]

x_i \ x_j	1	2	3	4	5	6	7	8	9
1	–	28	21	14	17	18	22	15	30
2	28	–	47	36	25	20	35	38	50
3	21	47	–	26	37	30	20	13	18
4	14	36	26	–	15	31	34	25	17
5	17	25	37	15	–	29	16	19	45
7	18	20	30	31	29	–	16	19	45
8	15	38	13	25	22	19	12	–	28
9	30	50	18	17	35	45	32	28	–

이 예제에서 하한값(LBB1 과 LB2)의 계산 절차는 생략하고, 나무탐색 알고리듬 1과 2는 하한값을 구하는 방법만 다를 뿐 나머지는 모두 같기 때문에 알고리듬 2에 대해서만 설명하기로 하고, 결과에 대해서는 두 알고리듬 모두 보여주기로 한다.



[그림 2] 지점(고객)의 지리적 그래프



[그림 3] 하한값 LB1과 알고리듬을 이용한 예제의 분단탐색나무

4.1 알고리듬 1(하한값 LB1을 이용)

[그림 3]은 앞에서 언급했듯이 자세한 과정을 생략하고, 알고리듬 1을 이용하여 문제를 해결한 결과를 나타내주고 있다. 분단지점을 나타내는 원내의 숫자는 그 분단지점에서 분단을 위해 선택된 지점(고객)을 의미한다. 숫자는 그 분단지점에 상응하는 단계까지의 루트를 확장하기 위해 선택되었던 지점이 제외된 것을 의미한다.

분단지점을 나타내는 원위 왼쪽에 있는 숫자는 계산된 하한값(LB1)을 나타낸다.

4.2 알고리듬 2(하한값LB2를 이용)

여기서는 알고리듬이 전개되는 모든 절차(하한값 계산절차 제외)를 기술하여 자세히 설명하기로 한다.

[step 1] : (초기값)

L=0, N=0 으로 설정.

$Z_{opt}=204.0$; 이 값은 3.3 항에서 (2)를 이용하여 얻은 값이다.

즉 ZL*1.04=203.1---> 204.0

$Z(0)=LB_0=195.34$(최초 분단지점에서의 최선의 하한값)

[step 2] : (분단을 위해 가능한 고객의 집합이 존재하는지 검토).

이 단계에서 가능한 고객의 집합이 존재하므로, GOTO [step 3].

[step 3] : (분단)

고객 x_4 가 분단지점 1에서 분단을 위해 선택된다.

$N=N+1=0+1=1$, 그리고

$L=L+1=0+1=1$

다음에, GOTO [step 4].

[step 4] : (분단지점 N에서의 분단과 비용행렬 수정)

분단지점 1에서 분단을 실시하고, 지금까지 부분적으로 완성된 루트 (x_1-x_4)에 대한 비용행렬을 수정한 다음, GOTO [step 5].

[step 5] : (하한값의 계산)

$Z(1)=LB1=195.34$ 그리고 GOTO [step 6~9]

[step 6~9] : (전진분단 또는 후진추적 여부 결정)

Z(1)이 실현가능해를 도출하지 않았으며, $Z(1) < Z_{opt}$ 이므로 GOTO [step 2].

[step 2] : 분단을 위한 고객이 아직까지 존재하므로, GOTO [step 3].

[step 3] : 지금까지 부분적으로 완성된 루트 (x_1,x_4)를 확장하기 위하여 또 하나의 고객을 선택한다. 분단지점 2 로 전진분단하기 위하여 고객 x_5 가 선택된다.

$N=N+1=1+1=2$, 그리고

$L=L+1=1+1=2$

다음에 GOTO [step 4].

[step 4] : 분단지점 2에서 분단을 실시하고, 지금까지 부분적으로 완성된 루트 $(x_1-x_4-x_5)$에 대한 비용행렬을 다음과 같이 수정한다.

$c_{i4}=c_{4i}=\infty$ ($i=2,\cdots,9$에 대하여), 그리고

$c_{45}=c_{54}=15$

다음에 GOTO [step 5].

.

.

.

[step 5] : $Z(8)=LB7=199.0$

다음에 GOTO [step 6~9].

[step 6~9] : Z(8)은 실현가능해를 도출해 냈으며, $Z(8) < Z_{opt}$ 이므로 새로운 $Z_{opt}=Z(8)=199.0$ 으로한다. 그리고 이 해의 값에 해당하는 루트들을 기록한다.

다음에 GOTO [step 10](역추적분단 실시를 위해).

여기서 부분문제들에 대한 현재의 실현가능해의 값(199.0)에 상응하는 q-루트들은 다음과 같다.

Route 1 : $x_1 - x_6 - x_2 - x_3 - x_1$

Route 2 : $x_1 - x_8 - x_7 - x_1$

Route 3 : $x_1 - x_4 - x_5 - x_9 - x_1$

[step 10] : 분단대안이 검토되지 않았으므로, GOTO [step 10a].

[step 10a] : $N=N+1=8+1=9$

다음에 분단지점 9로 역추적분단을 실시하고, 비용행렬을 수정한다. 그리고 GOTO [step 5]

.
.
.

[step 5] : Z(14) = LB14 = 202.0
그리고 GOTO [step 6~9].

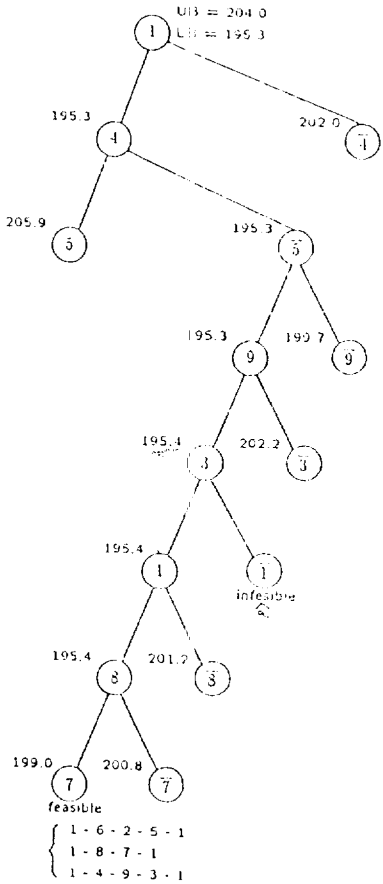
[step 6~9] : Z(14)는 실현가능해를 도출하지 않았으며, Z(14) 〉 Zopt 이므로 GOTO [step 10] (역추적 분단을 위해).

[step 10] : 분단대안이 이미 검토되었으므로 GOTO [step 10b].

[step 10b] : L = L − 1 = 1 − 1 = 0,
이에 상응하는 비용행렬 수정 다음에 GOTO [step 10c].

[step 10c] : L = 0 이므로 알고리듬의 모든 절차를 끝낸다.

이와같은 절차를 끝내므로 인하여 알고리듬 2는 14번째 분단지점에서 끝나게 되며, 이때 최적해의 값은 199.0 이 되었다. 이와같은 차량경로문제의 '예'에 대한 완전한 나무탐색의 모형은 [그림 4]와 같다.



[그림 4] 한계값 LB2와 알고리듬을 이용한 예제의 분담탐색 나무 모형도

5. 계산결과

지금까지 차량경로 문제에 대한 한계값을 구하는 방법과 나무탐색알고리듬을 이용하여 최적해를 구하는 절차를 보여주었다. 여기서는 이와 같은 알고리듬과 두가지 한계값(LB1과 LB2)을 이용하여 여러가지 예를 가지고 실험한 계산결과를 제시하고 다른 알고리듬에 의한 결과와 비교해 보기로 한다. 여기서 사용한 테스트 문제 는 N=9부터 40까지 지점(분배센터+고객)을 갖는 문제들이다. 다음 도표에서 보는 바와 같이 문제들 중의 문제 2에서 문제 7까지는 문헌 [11], [6], [20]에 나타난 것이며, 문제 1은 임의로 만든 문제이다.

〈표 2〉 테스트 문제

문 제	지점의 수	총 수요량	차 량 수	적 재 량	출 처
1	9	15	3	6	주어진 예제
2	11	93	4	24	문헌[11]의 문제
3	16	258	5	55	〃
4	16	258	3	90	〃
5	21	329	6	58	〃
6	21	329	4	85	〃
7	40	140	5	30	문헌 [20]의 문제

〈표 2〉에서는 문제의 크기와 출처, 그리고 〈표 3〉에서는 문제에 대한 최적해의 값, 최초 하한값(LB1과 LB2) 그리고 각 알고리듬에 대한 분단의 수를 보여주고 있다. 여기서 사용한 컴퓨터의 기종은 퍼스널 컴퓨터(IBM, PS/2 70-386)이며 프로그램실행은 마이크로소프트 포트란 4.0 컴파일러를 이용하였다. 또한 〈표 3〉에서는 본 연구에서 개발한 알고리듬 1, 2의 결과와 문헌에 나타난 알고리듬[13]의 결과와 비교하고 있다.

〈표 3〉 계산결과 및 비교

문제	최적해	최초하한값				분단지점 수			
		알고리듬1	a	알고리듬2	b	알고리듬1	a	알고리듬2	b
1	199.0	198.7	·	195.3	·	14	·	14	·
2	222.7	222.6	211.0	222.7*	222.7*	8	49	0	0
3	334.1	323.9	298.1	351.1	321.4	536	3,336	86	194
4	277.9	266.9	252.1	267.8	265.5	252	2,148	188	498
5	429.9	413.6	381.2	429.9*	429.7	1.382	–	0	6
6	357.6	341.5	346.8	346.4	–	–	208	886	
7	783.0	713.4	·	767.9	·	–	·	550	·

a 와 b : 문헌 [6]에서의 알고리듬
* : 최초한계값을 구하는 과정에서 얻어진 최적해
· : 알고리듬 a와 b에 의해 테스트되지 못한 경우
– : 해를 얻지 못한 경우

〈표 3〉의 결과에서 보듯이, 여기서 사용된 간접하한값(LB2)은 최적해의 평균 1.86%내에 있으며, 직접하한값(LB1)의 경우는 평균 3.62% 내에 있다. 이와 같은 한계값을 사용하여 나무탐색 알고리듬을 조금만 더 개선한다면 보다 큰 문제를 해결 할 수 있다는 확신을 가지게 한다. 또한 최적해를 얻기 위해 계산과정에서 요구되는 최초 하한값(LB1과 LB2) 및 분단지점의 수가 모두 기존의 문헌에서 나타난 결과보다 상당히 좋은 결과를 얻었다는 것이다. 마지막으로 여기서 개발된 알고리듬 2에 의해 해결된 40개 지점의 차량경로 문제는 어떤 문헌에서 해결한 문제규모(최적해를 얻는 경우)보다 크다고 할 수 있다.

6. 결 론

차량경로문제(VRP)는 조합최적화 문제의 분야에서는 이론적으로나 실제에서 모두 중요한 가치를 갖는 분야이다. 이 분야의 문제는 지난 30여년간 상당히 심층적으로 연구되어왔으며, 그 결과 이 문제를 주제로 하여 방대한 양의 문헌이 나와 있다. 비록 최적해를 구하기 위한 알고리듬에 관한 연구는 이 기간중 계속적으로 발전을 해 왔음에도 불구하고 해결할 수 있는 문제의 규모에 있어서는 크게 발전하지 못하였다. 즉, 10여년 전쯤에 25개의 지점을 갖는 차량경로문제의 최적해를 구할 수 있었음에도 불구하고, 지금까지 30개 지점을 갖는 문제의 최적해를 구할 수 있는 정도까지 밖에 발전하지 못하였다. 물론 아주 특정한 경우, 보다 쉬운

방법을 이용하여 제한적이나마 100여개의 지점을 갖는 문제를 해결할 수는 있었지만, 이것은 모든 경우의 문제를 해결할 수 있는 일반적인 알고리듬이 아닌, 특정조건을 부가해서 얻어진 경우에 불과하다.

본 연구는 나무탐색법(tree-search method)과 상태공간 완화법으로부터 얻은 한계값을 이용하는 것을 기초로 하여 차량경로문제의 최적해를 구하는 알고리듬을 개발하였다. 그 결과 알고리듬 2의 경우 문헌상에 나타난 차량경로문제의 최적해를 구하는 어떤 방법보다 좋은 방법이 된다고 생각할 수 있게 되었다. 또한 나무탐색법을 이용한 알고리듬의 개발로 이제까지 문헌상에서 볼 수 없었던 40개 지점의 규모를 갖는 일반적인 차량경로문제에 대한 최적해를 구할 수 있었다는 것이다. 알고리듬을 개발하는 과정에서 분단전략(branching strategy)을 다양하게 적용하여 완전한 알고리듬으로 개선하려는 노력을 시도하였으며, 이러한 노력이 더 이루어진다면 보다 좋은 결과가 얻어질 수 있다는 것을 염두에 두고, 이 점이 앞으로 더 연구되어야 할 과제로 제시된다.

마지막으로 이 연구의 결과 얻어진 방법론은 현재까지 현실문제에 많은 적용이 되고 있지는 않지만, 군사분야의 병참수송문제등과 산업분야에서의 수송, 분배, 생산라인 및 전자분야의 회로문제 등 다양한 분야에서의 적용이 가능할 것으로 생각된다.

참 고 문 헌

[1] Balas, E. and P. Toth, "Branch and Bound Methods," in Traveling Salesman Problem, E. L. Lawler et. al. Eds. , 1985, pp. 361-401.

[2] Balinski, M. L. and R. E. Quandt, "On an Integer Program for a Delivery Problem," *Operations Research*, vol. 12, 1964, pp. 300-304.

[3] Bellman, R. , *Dynamic Programming*, Princeton Univ. Press, Princeton, 1958.

[4] Christofides, N. and S. Eilon, "An Algorithm for the Vehicle Dispatching Problem," *Operational Research Quarterly*, vol. 20, 1969, pp. 309-318.

[5] Christofides, N. , "The Traveling Salesman Problem," in *Combinatorial Optimization*, N. Christofides, A. Mingozzi, P. Toth and C. Sandi, Eds, Wiley, Chichester, 1979.

[6] Christofides, N. , A. Mingozzi and P. Toth, "Exact Algorithms for the Vehicle Routing Problem based on Spanning Trees and Shortest Path Relaxation," *Mathematical Programming*, Vol. 20, 1981, pp. 255-282.

[7] Christofides, N. , A. Mingozzi and P. Toth, *State-Space Relaxations for Combinatorial Problems*, Report IC-OR-79-09, July, Imperial Collage of Science and Technology, London, 1979.

[8] Christofides, N. , A. Mingozzi and P. Toth, "State-Space Relaxations Procedures for the Computation of Bounds to Routing Problems," *Networks*, Vol. 11, 1981, pp. 145-164.

[9] Clarke, G. and J. W. Wright, "Scheduling of Vehicle from a Central Depot to a Number of Delivery Points," *Opera-*

tions Research, Vol. 12, 1964, pp. 568.

[10] Danzig, G. B. and K. H. Ramser, "The Truck Dispatching Problem," *Operations Reaearch*. Vol. 12, 1959, pp. 80.

[11] Eilon, S. , C. Watson-Gandy and N. Christofides, *Distribution Management : Mathematical Modeling and Practical Analysis*, Griffin, London, 1971.

[12] Fisher, M. , *Lagrangean Relaxation Methods for Combinatorial Optimization*, Paper Presented at Summer School in Combinatorial Optimization, Urbino, Italy, 1978.

[13] Garfinkel, R. and G. Nemhauser, *Integer Programming*, Wiley, New York, 1970

[14] Gaskel, T. J. , "Bases for Vehicle Fleet Scheduling," *Operational Research Quarterly*, Vol. 18, 1967, pp. 281.

[15] Geoffrion, A. , "Lagrangean Relaxation and It's Uses in Integer Programming," *Mathematical Programming Study*, Vol. 2, 1974, pp. 82-114.

[16] Gillet, B. E. and L. R. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem," *Operations Research*, Vol. 22, 1974, pp. 340.

[17] Hays, R. , The Delivery Problem, *Management Science Research Report*, No. 106, Carnegie Institution of Technology, 1967.

[18] Held, M. , R. M. Karp and H. P. Crowder, "Validation of Subgradient Optimization," *Mathematical Programming*, Vol. 6, 1974, pp. 62-88.

[19] Lawler, E. , J. Lenstra, A. Rinnooy Kan and D. Shmoys, *Traveling Salesman Problem*. Wiley, Chichester, 1987.

[20] Lee, K. H. , "Algorithms for Routing Problems in Distribution," Ph. D. Thesis, Dept. of Management Science, Imperial Collage, 1991.

[21] Lucena Filho, A. P. , "Exact Solution Approaches for the Vehicle Routing Problems," Ph. D. Thesis, Dept. of Management Science, Imperial Colage, 1986.

[22] Pierce, J. F. , *A Two Stage Approaches for the Solution of Vehicle Dispatching Problems*. Presented at 17th TIMS International Conference, London, 1970.

[23] Sandi, C. , "Subgradient Optimization," in *Combinatorial Optimization. N* Christofides, A. Mingozzi, P. Toth and C. Sandi, Eds. , Wiley, Chichester, 1979, pp. 73-91.

[24] Tillman, F. A. and H. Cochran, "A Heuristic Approach for Solving the Delivery Problem," *Journal of Industrial Engineering*, Vol. 19, 1969, pp. 354.

[25] Waters, C. D. J. and G. P. Bodie, "Realistic Size for Routing Problems," *Journal of Operational Research Society*, Vol. 38, 1987, pp. 565-566.