

最適車輛運行을 爲한 分枝限界技法

(A Branch-and-Bound Algorithm for the Optimal Vehicle Routing)

宋 省 憲*
朴 淳 達*

ABSTRACT

This study is concerned with the problem of routing vehicles stationed at a central depot to supply customers with known demands, in such a way as to minimize the total distance travelled.

The problem is referred to as the vehicle routing problem and is a generalization of the multiple traveling salesmen problem that has many practical applications.

A branch-and-bound algorithm for the exact solution of the vehicle routing problem is presented.

The algorithm finds the optimal number of vehicles as well as the minimum distance routes. A numerical example is given.

1. 序 論

병참시스템의 주요요소의 하나는 재화나 서비스를 정규적으로 수집·배분할 목적으로 차량을 할당하고 운행하는 것이다. 예를 들면 신문배달 [2], 도시 쓰레기 수거 [4], 통근 버스 운행, 연료배달, 기타 산업에서의 트럭 운행 등이 있다. [1]

이러한 시스템은 창고가 한 개일 수도 있고, 여러개 일 수도 있다. 목적은 비용최소화(배분비용, 차량 및 창고구입 비용등)이거나 서비스 개선(분배시간 단축등)에 있다. 제약조건은 다음 항목에 관계될 수 있다.

- i) 창고 (수, 가능한 위치, 수용능력)
- ii) 차량 (형태, 수, 수용능력)
- iii) 배달지점 (수요량, 배달 도착시간)
- iv) 운행조건 (최대운행시간, 최대운행거리, 도로의 수용능력)
- v) 작업자 시간계획 및 할당 (노조규약)
- vi) 시스템 동태 (재고보유, 분배 및 지연 시간)

본 연구에서는 한개의 고정된 창고에 수용능력이 일정한 여러대의 차량이 있고, 배달지점의 수요량을 미리 알고, 차량의 최대운행시간이 정해진 항목들을 제약조건으로 할 때, 창고에 대기하고 있는 이들 차량으로 모든 지점

의 수요량을 만족시키면서 총 운행거리를 최소화하기 위한 배달경로를 찾는 문제를 고려한다.

대부분의 차량운행 모형의 가장 단순한 기본 형태는 외판원 문제 (Traveling Salesman Problem ; TSP)이다. 외판원 문제에 판매원수가 여러명으로 되면 복수 외판원 문제 (Multiple Traveling Salesmen Problem ; MTSP)가 된다. 복수 외판원 문제는 네트워크 (net work)에 N개의 노드 (nodes)가 있고, M명의 판매원이 있을때, 각 노드(출발지 제외)가 한명의 판매원에 의해 한번만 방문되고, M명의 판매원에 의한 총 여행거리 (비용)을 최소화하는 M개의 subtours를 찾는 것이다. MTSP를 TSP모형으로 변형시키는 방법이 몇편의 논문[3], [10]에서 제시되었다.

차량운행문제 (Vehicle Routing Problem ; VRP)는 복수 외판원 문제에 차량수용 능력 (vehicle capacity), 최대운행시간 등의 여러 제약조건을 첨가한 일반화된 문제이다. 이러한 문제는 Dantzig 와 Ramser[7]에 의해 최초로 고려되었는데, 그는 선형계획법 (linear programming)을 이용한 발견적 기법 (heuristic algorithm)을 제시하였다. Clarke와 Wright[6]은 “절약”(saving) 개념을 도입한 발견적 기법을 개발하였다. 예를들어 설명하면, 처음에는 그림 2.1와 같이 2개의 수요지점 i와 j가 2대의 차량으로 각각 공급받는다고 하자. 지점 i와 지점 j를 그림 2.2와 같이 연결하여 2대의 차량대신에 1대의 차량으로 운행한다면, 운행거리에 있어서 $(2d_{oi} + 2d_{oj}) - (d_{oi} + d_{oj} + d_{ij}) = d_{oi} + d_{oj} - d_{ij}$ 의 절약 (saving)이 생긴다.

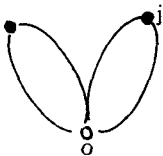


그림 2.1 초기상태

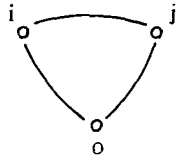


그림 2.2 지점 i와 j를 연결한 상태

이 공식에 의거 수요지점 (demand points) 간의 “절약 (saving)”을 계산하여 크기 순으로 절약이 큰 것부터 차례로 열거하여 목록 (list)을 만든다.

현재의 최대절약을 나타내 주는 지점 l 과 지점 j에 대해서, 이용 가능한 트럭의 수요능력 조건을 넘지 않는 범위에서 두 지점을 연결해 나간다. 이 방법은 추후 개발된 대부분의 발견적 기법의 기본이 되었다.

이 Clarke -Wright 방법에 몇가지 제약조건을 첨가하고 수정한 기법이 상용 프로그램에 사용되고 있다. 그런데 Gaskell 은 이러한 프로그램에 의한 해가 종종 최적해와는 차이가 큼을 보였다. 최적해 (optimal solution)를 주는 정확한 기법 (exact algorithm)으로서, Foster 와 Ryan[8]은 수정 단체법 (revised simplex method)을 사용한 정수 계획법적 접근방법 (integer programming approach)를 제시했으며, Christofides, et al [5]은 동적계획법 (dynamic programming)을 이용한 나무형 탐색 (tree search) 기법을 제시하였다.

Clarke -Wright 방법같은 발견적 기법들은 tour를 형성해 나가기 때문에 필요한 차량대수 또는 경로 (routes)수가 부수적으로 산출되지만, 앞에서 기술된 정확한 기법들은 차량의 대수를 미리 지정해 놓고, 그 정해진 차량대수 또는 경로수에 대해서 총 운행거리를 최소화하기 때문에, 이를 위한 적정차량대수 또는 경로수를 산출하지는 않는다.

따라서 본 연구에서는 총 운행거리를 최소화하면서 적정차량대수를 산출하기 위한 정확한 기법 (exact algorithm)으로서 분枝限界技法 (Branch -and -Bound Algorithm)을 제시한다.

2 . 分枝限界技法

2-1 수식모형

본 연구에서 취급되는 차량운행문제의 정수

계획 수식모형 (integer programming formulation) 은 아래 (P) 와 같다.[5] [9]

N ; 노드수

M ; 차량수

P ; 차량의 수용능력

T ; 각 경로상에서의 최대운행 시간

Q_i ; 노드 i 에서의 수요량 ($Q_1 = 0$)

t_{ij} ; 노드 i 에서 노드 j 까지의 최소시간 (least time)

U_i ; 노드 i 에서 수요량 Q_i 를 하역하는데 걸리는 시간

d_{ij} ; 노드 i 에서 노드 j 까지의 최단거리 (또는 “비용”)

$$X_{ijk} = \begin{cases} 1, & \text{arc}(i, j) \text{가 차량 } k \text{에 의해} \\ & \text{방문될 때} \\ 0, & \text{그렇지 않을 때} \end{cases}$$

식 (2-1) 은 총거리 (또는 “비용”) 를 최소화하는 것이다.

식 (2-1), (2-3) 은 각 수요지점 (demand node) 이 한대의 차량으로 처리됨을 표시한다. 식 (2-4) 는 차량이 한 수요지점을 방문하면 또한 반드시 떠남을 표시한다. 식 (2-5), (2-6) 은 각 경로(route) 에 대한 차량 수용능력과 최대운행시간의 조건식이다. 식 (2-7), (2-8) 은 차량의 이용가능성 (vehicle availability) 이 초과되지 않음을 표시한다.

(P)

$$\text{Minimize } Z = \sum_{i=1}^N \sum_{j=1}^N (d_{ij} \sum_{k=1}^M X_{ijk}) \quad (2-1)$$

$$\text{Subject to } \sum_{i=1}^N \sum_{k=1}^M X_{ijk} = 1 \quad (j = 2, \dots, N) \quad (2-2)$$

$$\sum_{j=1}^N \sum_{k=1}^M X_{ijk} = 1 \quad (i = 2, \dots, N) \quad (2-3)$$

$$\sum_{i=1}^N X_{ipk} - \sum_{j=1}^N X_{pjk} = 0 \quad \begin{matrix} (K = 1, \dots, M) \\ (P = 1, \dots, N) \end{matrix} \quad (2-4)$$

$$\sum_{i=1}^N (Q_i \sum_{j=1}^N X_{ijk}) \leq P \quad (K = 1, \dots, M) \quad (2-5)$$

$$\sum_{i=1}^N \sum_{j=1}^N t_{ij} X_{ijk} + \sum_{i=1}^N (U_i \sum_{j=1}^N X_{ijk}) \leq T \quad (K = 1, \dots, M) \quad (2-6)$$

$$\sum_{j=2}^N X_{1jk} \leq 1 \quad (K = 1, \dots, M) \quad (2-7)$$

$$\sum_{i=2}^N X_{i1k} \leq 1 \quad (K = 1, \dots, M) \quad (2-8)$$

$$Y_i - Y_j + N \sum_{k=1}^M X_{ijk} \leq N - 1 \quad (i \neq j = 2, \dots, N) \quad (2-9)$$

$$X_{ijk} = 0 \text{ or } 1 \text{ for all } i, j, k \quad (2-10)$$

Y_i 임의의 실수

식 (2-9)는 외판원 문제에서 subtour를 제거하여 각 경로가 본점을 통과하도록 Miller, Tucker, Zemlin이 유도한 조건식이다. 식 (2-10)은 정수조건이다.

그런데 식 (2-2)와 (2-4)는 (2-3)을 내포하고, 식 (2-4)와 (2-7)은 (2-8)을 내포하므로, 보통 식 (2-3)과 (2-8)을 제외한다. 이 모형에서 수요지점이 방문되면 각 수요량이 만족된다는 가정을 세웠다. 그런데 위 수식은 너무나 복잡해서, 작지 않은 크기(non-trivial size)의 차량운행 문제를 풀기가 힘들다는 것은 명백하다.

2-2. 그래프에서의 가능해

그래프 $G(N, A)$ 에서 $r(1 \leq r \leq M)$ 개의 경로(routes)가 다음 사항을 만족하면 차량운행 문제의 한 가능해(a feasible solution)가 된다.

- (i) 각 경로는 노드 1을 출발점 및 도착점으로 한다.
- (ii) 노드 1을 제외한 그래프의 모든 노드는 r 개의 경로중 하나에 속한다.
- (iii) 각 경로는 차량의 수용능력과 최대 운행

시간조건을 만족한다. 이 r 개의 경로에 의한 가능해를 “ r -tour”라고 정의하자.

2-3. 그래프의 확대

M 개의 차량이 각각 다른 곳에서 출발한다고 가정하면 $M-1$ 개의 가상출발점이 추가되어, 아래와 같이 확대된 그래프 $G(N', A')$ 가 형성된다. 즉 노드집합 N' 는 $N+M-1$ 개의 노드로 구성되며, 출발점의 노드번호는 1번부터 M 번까지가 된다. arcs의 집합 A' 는 다음 4가지 항목으로 구성된다.

- (i) A 의 모든 arc
- (ii) arc $(i, j+M-1)$, $i=1, \dots, M$,
 $j=2, \dots, N$
- (iii) arc $(i+M-1, j)$ $i=2, \dots, N$
 $j=1, \dots, M$
- (iv) arc (i, j) , $i=1, \dots, M$,
 $j=1, \dots, M$

그래프 $G(N', A')$ 에서 각 arc의 거리 $d(i, j)$, 수요량 $Q'(i)$, 노드 i 에서의 소요시간 $U'(i)$, 두 지점간의 운행시간 $t'(i, j)$ 는 다음과 같다.

$$d'(i+M-1, j+M-1) = d(i, j)$$

$$d'(i, j+M-1) = d(1, j)$$

$$d'(i+M-1, j) = d(i, 1)$$

$$d'(i, j) = \begin{cases} 0 \\ \infty \\ \infty \end{cases}$$

$$Q'(i+M-1) = Q(i)$$

$$Q'(i) = Q(1)$$

$$U'(i+M-1) = U(i)$$

$$U'(i) = U(1)$$

$$t'(i+M-1, j+M-1) = t(i, j)$$

$$t'(i, j+M-1) = t(1, j)$$

$$t'(i+M-1, j) = t(i, 1)$$

$$t'(i, j) = 0$$

$$i=2, \dots, N, j=2, \dots, N \quad (2-13)$$

$$i=2, \dots, M, j=2, \dots, N \quad (2-14)$$

$$i=2, \dots, N, j=1, \dots, M \quad (2-15)$$

$$j=i-1, i=MM+1, \dots, M$$

$$j=i-1, i=1, \dots, MM \quad (2-16)$$

$$j \neq i-1, i=1, \dots, M, j=1, \dots, M$$

$$i=2, \dots, N \quad (2-17)$$

$$i=1, \dots, N \quad (2-18)$$

$$i=2, \dots, N \quad (2-19)$$

$$i=1, \dots, M \quad (2-20)$$

$$i=2, \dots, N, j=2, \dots, N \quad (2-21)$$

$$i=1, \dots, M, j=2, \dots, N \quad (2-22)$$

$$i=2, \dots, N, j=1, \dots, M \quad (2-23)$$

$$i=1, \dots, M, j=1, \dots, M \quad (2-24)$$

여기서 MM은 최소한의 필요차량수로서 아래와 같이 산출한다.

$$MM = \lceil \sum_{i=1}^N Q_i / P \rceil + 1$$

N ; 수요지점 (demand node) 의 수

Q_i ; 지점 i의 수요량
P ; 차량의 수요능력 (vehicle capacity)
위 수식의 이해를 돕기 위해 거리행렬을 그림으로 하면 2.1과 같다.

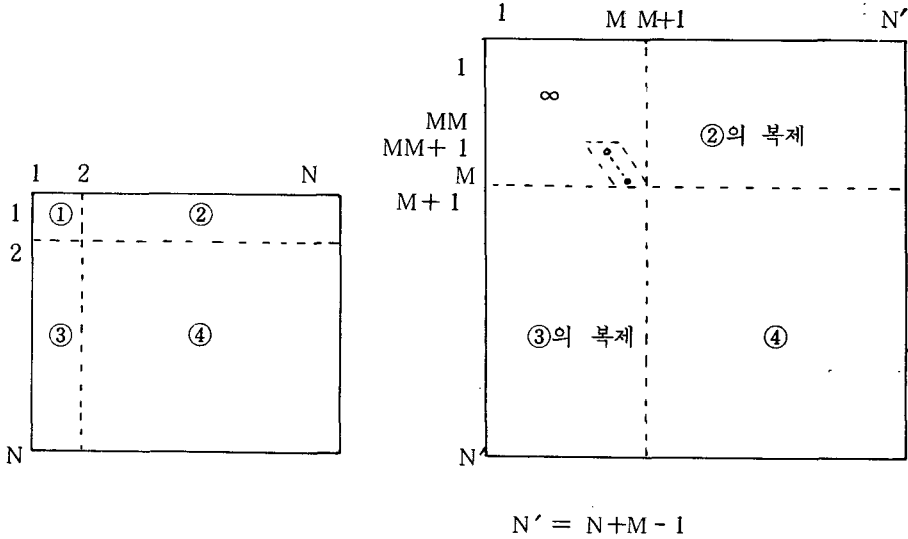


그림 2.1 원래 거리행렬과 확대된 거리행렬

확대된 그래프 $G(N', A')$ 에서 원문제 (P)는 제약조건식 (2.5)~(2.9)를 완화하면 아래 (P')와 같은 배정문제 (assignment problem)가 된다.[9]

(P')

$$\text{Minimize } Z' = \sum_{i=1}^{N'} \sum_{j=1}^{N'} d'_{ij} X_{ij} \quad (2-25)$$

$$\text{subject to } \sum_{i=1}^{N'} X_{ij} = 1, \quad j = 1, 2, \dots, N' \quad (2-26)$$

$$\sum_{j=1}^{N'} X_{ij} = 1, \quad i = 1, 2, \dots, N' \quad (2-27)$$

$$X_{ij} = 0 \text{ or } 1 \text{ for all } i, j$$

$$(2-28)$$

위 배정문제를 풀면 그 해가 r' ($1 \leq r' \leq M$)개의 subtours를 형성하며 차량운행문제의 하한값을 준다. 그런데 확대된 그래프 $G(N', A')$ 의 해는 반드시 1-tour일 필요는 없다. 확대된 그래프 $G(N', A')$ 에서 이들 r' 개의 subtours가 다음사항을 만족하면 차량운행 문제의 한 가능해가 되기 때문이다.

- (i) 각 subtour는 가상출발점 1, ..., M 중 적어도 하나를 포함한다.
- (ii) 가상출발점 1, ..., M을 제외한 모든 노드는 r' 개의 subtours중의 하나에 속한다.
- (iii) 각 subtour는 차량의 수용능력과 최대 운행시간 조건을 만족한다.

이는 그래프 $G(N, A)$ 의 r -tour와 같게 되는데, 이러한 개념은 Svestka와 Huckfe-

Idt[10]의 feasible tour, infeasible subtour, feasible subtour 개념과 같으며, 예를 들면 그림 2.2와 같다. 그림 2.2에서 $G(N', A')$ 의 (a), (e)는 꼭 같은 거리 Z 를 갖게 되며, 그래프를 축소시켜 $G(N, A)$ 로 환원

하면, $G(N', A')$ 의 1-tour(a)는 (b)로 되며, $G(N', A')$ 의 2-subtours (e)는 (f)가 되어 결국 (a)와 (e)는 $G(N, A)$ 에서 꼭 같은 경로를 갖는 2-tour가 된다.

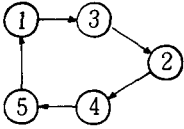
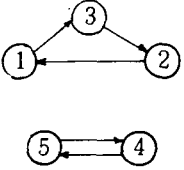
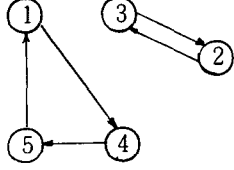
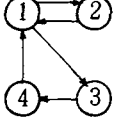
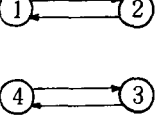
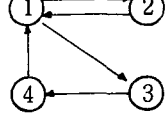
Graph	feasible tour	infeasible subtour	feasible subtour
$G(N', A')$	(a) 	(c) 	(e) 
$G(N, A)$	(b) 	(d) 	(f) 

그림 2.2 수요지점 4개, 차량 2대의 문제 예

2.4 分枝限界技法 (Branch-and-Bound Algorithm)

1 단계 : 그래프를 확대한다.

2 단계 : CLUB = 10^8 으로 놓는다.

CLUB : 차량운행문제의 최적해에 대한 현재의 최소 상한값.

(current least upper bound)

3 단계 : 관련된 배정문제를 푼다. 그해는 차량운행문제의 최적해의 하한값을 준다. 해의 가능성을 조사한다.

(check feasibility)

i) 가상 출발점 (1, ..., M)을 포함하지 않는 subtour가 있으면 (infeasible subtour, IFEAS = 3), 4 단계로 간다.

ii) subtours (또는 tour) 중 한개라도 차량 수용능력이나 최대운행시간조건을 만족하지 않으면 (infeasible subtour 또는 to-

ur, IFEAS = 2), 4 단계로 간다.

iii) 그렇지 않으면 (feasible subtours 또는 tour, IFEAS = 1), 배정문제의 최적해는 차량운행문제의 최적해가 된다.

10 단계로 간다.

4 단계 : arcs 수가 가장 적은 비가능해 (infeasible subtour)를 선택한다.

선택된 infeasible subtour의 arcs 수를 k 라 하자.

5 단계 : k 개의 subproblems로 가지를 친다. (branch) 즉 비가능한 subtour 또는 tour가 $I_1 \rightarrow I_2 \rightarrow \dots \rightarrow I_k \rightarrow I_1$ 이면, 다음과 같이 한다.

subproblem 1에 대해 $D(I_1, I_2) = \infty$

subproblem 2에 대해 $D(I_2, I_3) = \infty$

⋮

subproblem k 에 대해 $D(I_k, I_1) = \infty$

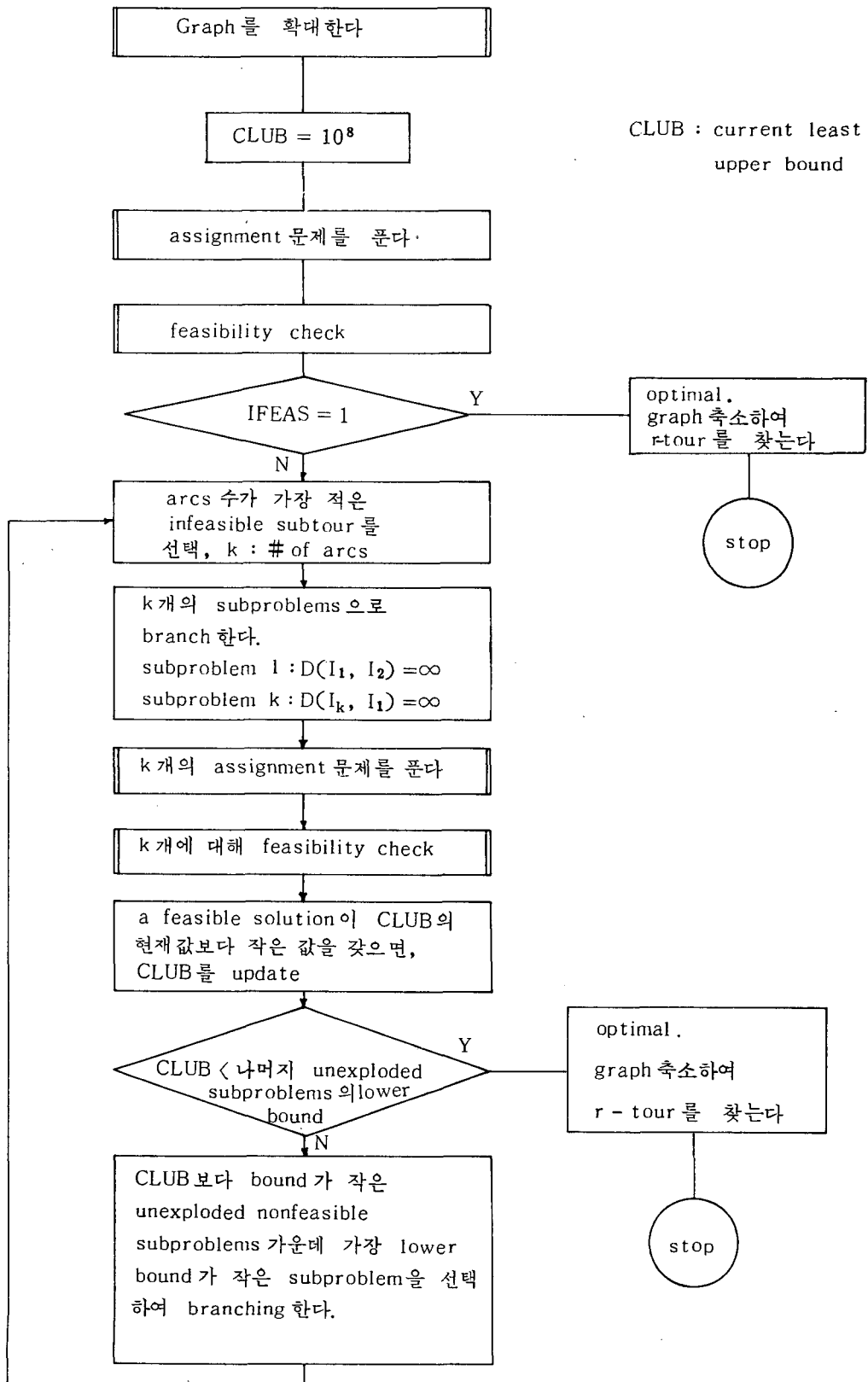


그림 2.3 차량운행문제에 대한 分枝限界技法의 흐름도

6 단계 : k 개의 새로운 배정문제를 푼다.
 각 해의 값 (solution distance) 는 해당 subproblem에 하한값이 된다.
 해의 가능성을 조사한다.
 (IFEAS = 1, 2, 3)

7 단계 : (i) 6 단계에서 가능해 한개 이상일 때, 이들 가능해의 가장 작은 거리 (Smallest Total Distance: STD) 가 CLUB보다 작으면, CLUB = STD로 놓고, 해당 가능해를 기록한다.

(ii) 그렇지 않으면, CLUB는 불변이다.

8 단계 : CLUB가 나머지 가지치지 않은 (unexploded) subproblems의 하한값보다 작으면, CLUB에 해당되는 해가 차량운행문제의 최적해가 되어, 10 단계로 간다. 그렇지 않으면 9 단계로 간다.

9 단계 : CLUB보다 하한값이 작은 가지를 차지 않은 비가능 (unexploded infeasible) subproblems 가운데서, 하한값이 가장 작은 subproblem을 선택하여 가지를 치기 위해 4 단계로 간다.

10 단계 : 확대된 그래프를 원래의 그래프로 축소환원하여 γ -tour를 구한다. 이 γ -tour가 차량 운행문제의 최적해가 되어 끝난다.

그림 2.3에 차량운행문제에 대한 분지한계 기법의 흐름도 (flow chart)가 묘사되어 있다.

3. 例題

어느 제조회사의 공장은 도시 1에 위치해 있는데, 도시 2에서 도시 9까지 8개의 도시에 1개씩있는 창고에 제품을 배달하려 한다. 공장은 현재 24톤 4대를 보유하고 있다. 각 도시간의 거리 및 수요량은 표3.1과

같다. 차량운행시간 및 배달시간에는 제한이 없다고 하자. 이때 최단거리운행 경로와 소요 차량수를 구하고자 한다.

이 문제를 풀기위해 우선 트럭 4대에 대해서 가상 출발점을 4개로 만들어 그래프를 확대하면 표3.2와 같게 된다. 표3.2의 거리도표를 이용해서 배정문제를 풀면 그해값이 차량운행문제의 하한값이 된다. 그림 3.1에서 보듯이 본 기법에 의한 최적해는 다음과 같다.

즉, 총거리 = 1582

운행경로

1 → 11 → 2 → 10 → 8 → 5 → 7 → 6 →

12 → 4 → 3 → 9 → 1

확대된 그래프를 축소하면 최적해는 다음과 같다.

총 운행거리 = 1582

운행경로

1 → 8 → 1

1 → 7 → 5 → 2 → 4 → 3 → 9 → 1

1 → 6 → 1

적정 차량수 = 3

4. 結論

여러 수요지점에 물건을 배달하는 차량운행 문제에 있어서 최단거리 운행경로는 물론 적정 차량수도 동시에 산출하는 분지한계기법을 개발하였다.

차량의 수용능력과 최대운행시간조건을 완화하면, 이 기법으로 복수 외판원 문제의 최적해를 구할 수 있으며 거기에서 차량수가 1대라고 한다면, 일반적인 외판원 문제 (Standard Traveling Salesman Problem)의 최적해를 구할수가 있다.

각 지점간 거리

출발지 \ 도착지	1	2	3	4	5	6	7	8	9
1	0	241	276	172	233	111	160	70	202
2	241	0	212	92	161	260	281	312	380
3	276	212	0	155	353	202	404	327	280
4	172	92	155	0	200	170	261	240	288
5	233	161	353	200	0	313	170	292	427
6	111	260	202	170	313	0	271	134	120
7	160	281	404	261	170	271	0	180	355
8	70	312	327	240	292	134	180	0	180
9	202	380	280	288	427	120	355	180	0

각 지점의 수요량

지 점	1	2	3	4	5	6	7	8	9
수 요 량	0	1	5	6	3	15	4	21	2

도표 3.1 각지점간 거리 및 수요량

각 지점간 거리

출발지 \ 도착지	1	2	3	4	5	6	7	8	9	10	11	12
1	99999	99999	99999	99999	241	276	172	233	111	160	70	202
2	99999	99999	99999	99999	241	276	172	233	111	160	70	202
3	99999	99999	99999	99999	241	276	172	233	111	160	70	202
4	99999	99999	0	99999	241	276	172	233	111	160	70	202
5	241	241	241	241	99999	212	92	161	260	281	312	380
6	276	276	276	276	212	99999	155	353	202	404	327	280
7	172	172	172	172	92	155	99999	200	170	261	240	288
8	233	233	233	233	161	353	200	99999	313	170	292	427
9	111	111	111	111	260	202	170	313	99999	271	134	120
10	160	160	160	160	281	404	261	170	271	99999	180	355
11	70	70	70	70	312	327	240	292	134	180	99999	180
12	202	202	202	202	380	280	288	427	120	355	180	99999

각 지점의 수요량

지 점	1	2	3	4	5	6	7	8	9	10	11	12
수 요 량	0	0	0	0	1	5	6	3	15	4	21	2

도표 3.2 확대된 그래프에서 각지점간 거리 및 수요량

CLUB = 10^8

$Z' = 1525$

배정문제의 목적함수값

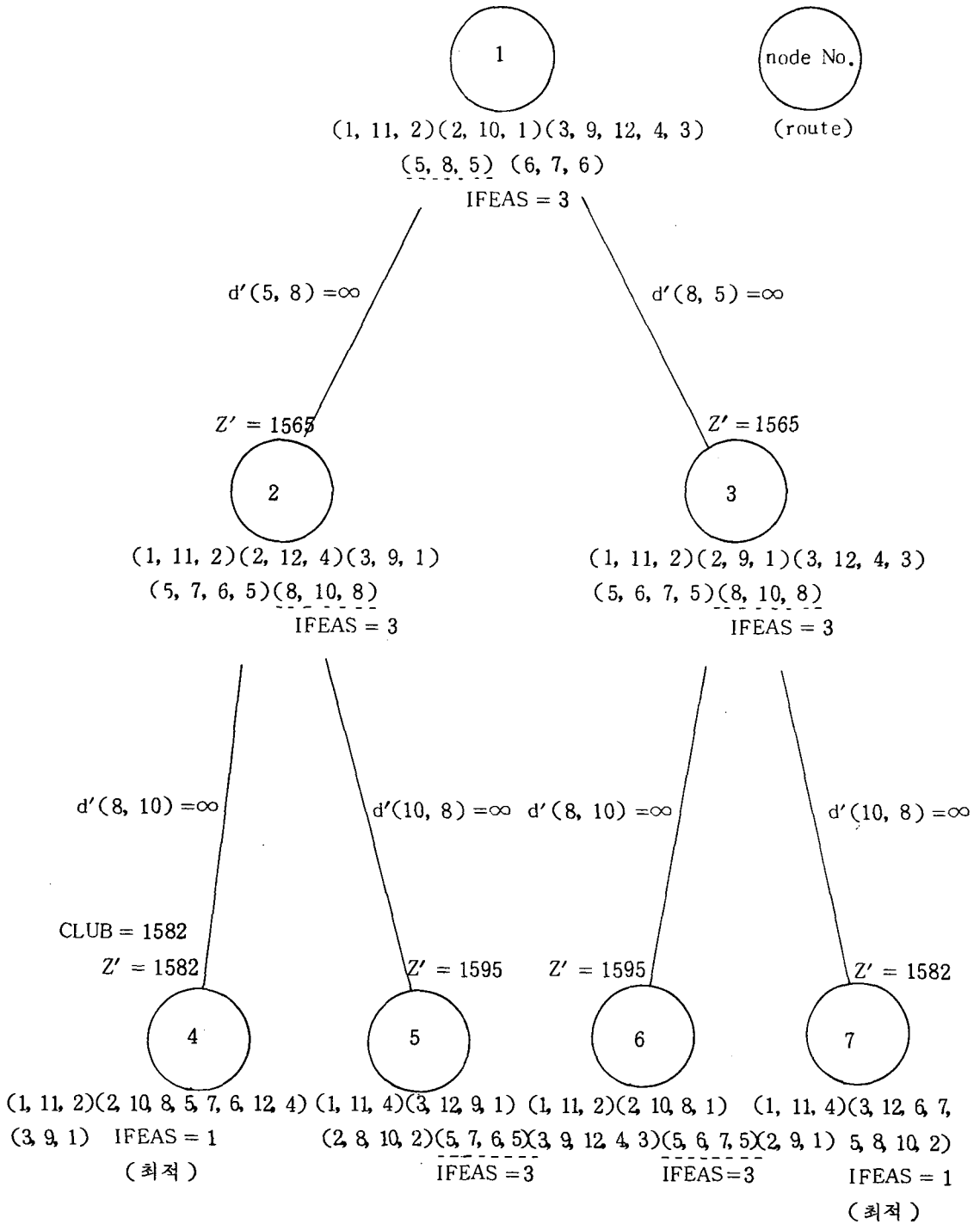


그림 3.1 解의 나무 (solution tree)

참 고 문 헌

宋省憲, 차량운행 문제에 대한 分枝限界技法의 개발, 서울大學校大學院, 1982. (석사 학위 논문)

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

Bellmore, M. and S. Hong, "Transformation of Multisalesman Problem to the Standard Traveling Salesman Problem," *J. of the Association for Computing Machinery*, Vol. 21, 1974, pp.500-504.

Beltrami, E.J. and L.D. Bodin, "Networks and Vehicle Routing for Municipal Waste Collection," *Networks*, Vol. 4, 1974, pp.65-94.

Christofides, N.A. Mingozzi and P. Toth, "Exact Algorithms for the Vehicle Routing Problem Based on Spanning Tree and

Shortest Path Relaxations," *Mathematical Programming (Netherlands)*, Vol. 20, No. 3, 1981, pp.255-282.

6. Clarke, G. and J.W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," *Operations Research*, Vol. 12, 1964, pp.568-581.
7. Dantzig, G.B. and J.H. Ramser, "The Truck Dispatching Problem," *Management Science*, Vol. 6, 1959, pp.80-91.
8. Foster, B.A. and D.M. Ryan, "An Integer Programming Approach to the Vehicle Scheduling Problem," *Operations Research Quarterly*, Vol. 27, 1976, pp.367-384.
9. Golden, B.L., T.L. Magnanti and H.Q. Nguyen, "Implementing Vehicle Routing Algorithms," *Networks*, Vol. 7, 1977, pp.113-148.
10. Svestka, J.A. and V.E. Huckfeldt, "Computational Experience with an M-Salesman Traveling Salesman Algorithm," *Management Science*, Vol. 19, 1972, pp.790-799.