

# 나무구조를 갖는 네트워크상에서의 제한용량이 있는 입지설정문제에 관한 연구\*

조 건†

전남대학교 경영학부

## A Study for a Capacitated Facility Location Problem on a Tree Structured Network

Geon Cho

School of Business Administration, Chonnam National University, Kwangju, 500-757

Given a tree structured network in which each node has its own demand and also stands for a candidate location of a potential facility, such as plant or warehouse, a capacitated facility location problem on the network (CFLPOT) is to decide capacitated facility locations so that the total demand occurred on the network can be satisfied from those facilities with the minimum cost. In this paper, we first introduce a mixed integer programming formulation for CFLPOT with two additional assumptions, the indivisible demand assumption and the contiguity assumption and then show that it can be reformulated as a tree partitioning problem with an exponential number of variables. We then show that it can be solved in  $O(n^2b)$  time by utilizing the limited column generation method developed by Shaw (1993), where  $n$  is the total number of nodes in the network and  $b$  is the maximum facility capacity. We also develop a depth-first dynamic programming algorithm with a running time of  $O(nb)$  for finding the locally maximal reduced cost which plays an important role in the limited column generation method. Finally, we implement our algorithms on a set of randomly generated problems and report the computational results.

**Keywords** : facility location problem, tree partitioning problem, dynamic programming, column generation method

### 1. 서론

입지설정문제(facility location problem)는 기업의 중요한 장기 의사결정문제 중의 하나로써 공장이나 창고와 같은 생산시설을 설치할 수 있는 후보입지가 주어지고 이러한 생산시설로부터 공급되는 제품에 대한 수요발생지역이 주어지며 또한 생산시설 설치비용 및 운영비용, 제품 수요량, 그리고 각 시설로부터 각 수요발생지역으로 제품을 수송할 때 소요되는 수송비용 등이 주어져 있다고 가정할 때, 최소의 비용으로 모든 제품 수

요량을 충족시키기 위한 생산시설들의 입지를 결정하는 문제이다. 이때 세우고자 하는 각 생산시설의 용량제한의 유무에 따라, 제한용량이 있는 입지설정문제(capacitated facility location problem : CFLP)와 제한용량이 없는 입지설정문제(uncapacitated facility location problem : UFLP)로 나누어진다.

일반적으로 CFLP는 정수계획 문제(integer programming problem)로 공식화되어 지고 NP-hard 문제이며(Mirchandani *et al.*, 1990), Lagrangian 휴리스틱을 사용함으로써 효율적으로 풀 수 있음은 잘 알려져 있다. 특히 Klineciewicz and Luss(1986)는 CFLP의 생산시설 용량에 대한 제약식(capacity constraints)

이 논문은 1998년 한국학술진흥재단의 학술연구비에 의하여 지원되었음.

† 연락처: 조건 교수, 500-757 광주광역시 북구 용봉동 300번지 전남대학교 경영학부, 경영연구소 상임연구원, Fax : 062-530-1449, e-mail : gcho@chonnam.ac.kr

2001년 1월 접수, 1회 수정 후, 2001년 5월 게재 확정.

을 완화하여 얻은 Lagrangian relaxation 문제(UFLP)에 대한 해를 활용한 알고리즘을 제시하였고, Barcelo and Casanovas (1984), Pirkul(1987), Sridharan(1993) 등은 수요량에 대한 제약식(demand constraints)을 완화하여 얻은 Lagrangian relaxation 문제(즉, 배낭문제(knapsack problem))에 대한 해를 활용한 알고리즘들을 제시하였다. 또한 Holmberg *et al.*(1999)은 분단탐색 구조(branch-and-bound framework) 내에서 subgradient 기법을 이용한 Lagrangian 휴리스틱과 matching 문제를 연속적으로 푸는 strong primal 휴리스틱을 동시에 활용한 새로운 해법을 제시하였다.

UFLP의 특별한 경우인 나무구조를 갖는 네트워크상에서의 제한용량이 없는 입지설정문제(uncapacitated facility location problem on a tree structured network : UFLPOT)는 다항식 시간 알고리즘(polynomial time algorithm)으로 풀 수 있는 반면 (Kolen, 1983; Cho, 1998), CFLP의 특별한 경우인 나무구조를 갖는 네트워크상에서의 제한용량이 있는 입지설정문제(capacitated facility location problem on a tree structured network : CFLPOT)는 일반적으로 CFLP와 마찬가지로 NP-hard 문제이다. CFLPOT는 목적함수의 유형 및 주어진 가정의 변화에 따라 많은 응용분야에서 다양한 형태로 발생되는데, 예컨대 통신분야의 근거리 통신망설계문제(local access telecommunication network design problem)(Shaw, 1993)나 경찰서, 소방서 등과 같은 공공설비나 백화점, 대형 할인매장, 자동차영업소 등과 같은 민간설비의 입지설정시 발생하는 나무상에서의 최대 적용범위 입지설정문제(maximum coverage location problem on a tree)(Megiddo *et al.*, 1983), 나무상에서의 그룹화된  $p$ -중심,  $p$ -미디안 문제(grouped  $p$ -center,  $p$ -median problems on a tree)(Averbakh and Berman, 1998)등을 들 수 있다. 본 연구에서는 특별히 CFLPOT에 두 가지 가정 [수요 비분리 가정(indivisible demand assumption)과 인접성 가정(contiguity assumption)]이 추가된 문제를 푸는 효율적인 알고리즘을 개발하고자 한다.

수요 비분리 가정은 각 지역에서 발생하는 수요량이 전량 단일 생산시설로부터 충족되어짐을 의미하며, 인접성 가정은 나무구조를 갖는 네트워크의 특성(즉, 임의의 두 지역을 연결하는 경로는 유일하다)을 잘 반영하는 매우 현실적인 가정으로써 한 지역에서 발생하는 수요량이 한 생산시설로부터 충족되어지는 경우 이 두 지역을 연결하는 경로상에 있는 지역에서 발생하는 모든 수요량도 동일한 생산시설로부터 충족되어짐을 의미한다. 이와 같은 두 가지 가정은 나무구조를 갖는 네트워크상에서의 통신 및 전력수송 집선장치(concentrator) 위치 선정문제, 전화국이나 소방서 입지설정문제, 파이프라인 시스템 설계문제 등에서 매우 중요하게 고려된다. 하지만 이러한 두 가지 가정하에서의 CFLPOT는 문제가 너무 제한적이라는 단점때문에 지금까지 연구자들의 연구대상에서 소외되어져 온 것으로 판단되며 결과적으로 문헌상에 발표된 이에 대한 연구결과는 전무한 것으로 판단된다. 따라서 본 연구에서는 먼저 위에서 언급한 두 가지 가정하에서의 CFLPOT에 대한

정수계획 모형을 소개한 다음, 이 두 가지 가정을 이용함으로써 CFLPOT가 지수적으로(exponentially) 많은 변수를 갖는 나무분할문제(tree partitioning problem)로 공식화될 수 있음을 보이고, 이는 Shaw(1993)가 제시한 제한된 열 생성기법(limited column generation method)의 핵심 아이디어를 확장·적용함으로써 모조다항식시간(pseudo-polynomial time)인  $O(n^2b)$ 시간으로 풀 수 있음을 보인다. 여기서  $n$ 은 주어진 네트워크상의 생산시설 설치가능 후보입지수(즉, 총노드수)를 나타내며,  $b$ 는 생산시설의 최대용량을 나타낸다. 또한 제한된 열 생성기법에서 중요하게 이용되는 소위 부분 최대감소비용(locally maximal reduced cost)은 제한용량이 있는 나무배낭문제(capacitated tree knapsack problem : CTKP)에 대한 최적값으로부터 얻어지는데, 이는 깊이우선 동적계획 알고리즘(depth-first dynamic programming algorithm)에 의해  $O(nb)$ 시간으로 구할 수 있음을 보인다.

본 논문은 다음과 같이 구성되어 진다. 2절에서는 CFLPOT의 정수계획 모형을 소개하고, 3절에서는 수요 비분리 가정과 인접성 가정을 이용하여 CFLPOT를 나무분할문제로 공식화할 수 있음을 보인다. 4절에서는 부분 최대감소비용의 개념을 정의하고 또한 이 비용은 깊이우선 동적계획 알고리즘에 의해  $O(nb)$ 시간으로 구할 수 있음을 보인다. 5절에서는 제한된 열 생성기법을 이용하여 CFLPOT를  $O(n^2b)$ 시간으로 풀 수 있음을 보이며, 6절에서는 본 연구에서 제시한 알고리즘을 C 언어로 코딩하여 여러 가지 유형의 문제에 대한 테스트를 실시한다. 마지막 7절은 결론으로 한다.

## 2. CFLPOT의 정수계획 모형

나무구조를 갖는 무방향 네트워크(undirected tree structured network)  $T = (V, E)$ 가 주어져 있다고 가정하자. 이때  $V = \{0, 1, 2, \dots, n\}$ 는 노드들(nodes)의 집합을 나타내며,  $E = \{(p_i, i) | i \in V \setminus \{0\}\}$ 는 호들(arcs)의 집합을 나타낸다. 여기서  $p_i$ 는 노드  $i$ 의 직상위 선행노드(predecessor node)로서 각 노드에 대해 유일하게 존재한다. 각 노드는 수요발생지역을 나타냄과 동시에 생산시설 설치후보지역을 나타내며, 각 호는 제품 수송로를 나타낸다. 본 연구에서는 편의상 각 노드들에 깊이우선탐색(depth first search : DFS)순서를 적용하여 0에서  $n$ 까지 번호를 부여하기로 한다. 이때 노드 0은 루트노드(root node)가 되며 명백히  $p_0$ 는 정의되지 않지만, 편의상  $p_0 = -1$ 이라고 가정한다. 또한 나무구조의 특성상 임의의 두 노드  $i, j (j < i)$ 를 연결하는 노드들의 집합을 경로(path)  $\gamma[i, j]$ (혹은  $\gamma[i, j]$ )라 할 때 이는 유일하게 존재하며,  $E$ 와  $V \setminus \{0\}$ 은 일대일 대응(one-to-one correspondence)관계에 있기 때문에 모든  $i \in V \setminus \{0\}$ 에 대하여 호  $(p_i, i)$ 를 간략히 호  $i$ 로 표시할 수 있다.

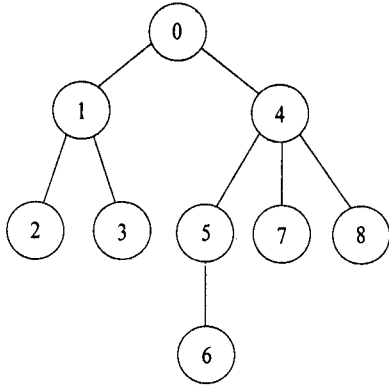


그림 1. 나무구조를 갖는 네트워크.

이제 임의의 두 노드  $i, j$ 에 대하여  $v^* = \max \{v \in V \mid v \in \gamma[j, 0] \cap \gamma[i, 0]\}$ ,  $\gamma[j, i] = \gamma[j, i] \setminus \{i\}$ ,  $\gamma[j, i] = \gamma[j, i] \setminus \{j\}$ 라고 하자. 만약 노드  $u$ 가  $\gamma[j, i]$ 상에 있다면, 노드  $i$ 에 대한 노드  $u$ 의 직상위 선행노드  $p_u^i$ 는 다음과 같이 정의된다.

- 1)  $u$ 가  $\gamma[j, v^*]$ 상에 있으면,  $p_u^i = p_u$ .
- 2)  $u$ 가  $\gamma[i, v^*]$ 상에 있으면,  $p_u^i = \{v \in \gamma[i, v^*] \mid p_v = u\}$  (즉,  $\gamma[i, v^*]$ 상에 있는  $u$ 의 직하위 후행노드(immediate successor))
- 3)  $p_u^i$ 는 정의되지 않으나, 편의상  $p_u^i = -1$ 로 한다.

예컨대, 다음에 주어진 <그림 1>에서  $i=3, j=6$ 이라고 할 때 경로  $\gamma[3, 6]$  상에 있는 노드 3에 대한 노드 0의 직상위 노드  $p_3^0$ 는 노드 1(즉,  $p_3^0 = 1$ )이고, 노드 3에 대한 노드 5의 직상위 노드  $p_3^5$ 는 노드 4(즉,  $p_3^5 = p_5 = 4$ )이다.

본 연구에서는 다음과 같은 두 가지 가정을 설정하고 아래에 소개된 몇 가지 기호(notation)들을 이용하여 CFLPOT를 정수계획 모형(CFLPOT)로 공식화(formulate)한다. 또한 3절에서도 볼 수 있듯이 다음의 두 가지 가정은 CFLPOT를 나무분할문제로 공식화하는 데 필수적인 역할을 한다.

- 1) 수요 비분리 가정(indivisible demand assumption) : 모든 노드에 대하여 각 노드의 총수요량은 한 생산시설로부터 전량 충족되어진다.
- 2) 인접성 가정(contiguity assumption) : 만약 노드  $j$ 의 총수요량이 노드  $i$ 에 위치한 생산시설로부터 충족되어진다면 경로  $\gamma[j, i]$ 상에 있는 모든 노드의 총수요량 또한 노드  $i$ 에 위치한 생산시설로부터 충족되어진다.

**기호 정의**

- $d_i$ : 노드  $i$ 의 총수요량
- $b_i$ : 노드  $i$ 에 위치한 생산시설 용량(일반적으로,  $b_i \geq d_i$ 라고 가정함)
- $c_i$ : 노드  $i$ 에 위치한 생산시설의 단위수요당 운영비용
- $f_i$ : 노드  $i$ 에 위치한 생산시설 설치비용

- $t_v$ : 호  $v$ 에 대한 단위수요의 단위거리당 수송비용
- $D_v$ : 호  $v$ 의 길이(거리)
- $c_{ij}$ : 노드  $i$ 에 위치한 생산시설로부터 노드  $j$ 의 총수요량을 충족시키는 데 소요되는 총비용(생산시설 운영비용 + 총수송비용)

$$x_{ij} = \begin{cases} 1, & \text{만약 노드 } i \text{에 있는 생산시설이 노드 } j \text{의 총수요를 충족시키면,} \\ 0, & \text{그렇지 않으면.} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{만약 노드 } i \text{에 생산시설이 설치되면,} \\ 0, & \text{그렇지 않으면.} \end{cases}$$

이때  $d_i, b_i, c_i, f_i, t_v, D_v$ 는 음이 아닌 정수들(nonnegative integers)로 가정하며  $x_{ij}, y_i$ 는 결정변수들(decision variables)을 나타낸다. 또한  $c_{ij}$ 는 다음과 같이 구해진다.

$$c_{ij} = \begin{cases} d_j(c_i + \sum_{v \in E_{ij}} t_v D_v), & \text{if } j \neq i \\ d_i c_i, & \text{otherwise.} \end{cases}$$

여기서  $E_{ij}$ 는 경로  $\gamma[i, j]$ 상에 있는 모든 호들의 집합을 나타낸다.

$$\text{(CFLPOT) } \min \sum_{i=0}^n \sum_{j=0}^n c_{ij} x_{ij} + \sum_{i=0}^n f_i y_i$$

$$\text{s.t. } \sum_{j=0}^n d_j x_{ij} \leq b_i y_i, \quad i = 0, 1, 2, \dots, n \quad (1)$$

$$\sum_{i=0}^n x_{ij} = 1, \quad j = 0, 1, 2, \dots, n \quad (2)$$

$$x_{ij} \leq x_{i p_j}, \quad i, j = 0, 1, 2, \dots, n \quad (3)$$

$$x_{ij}, y_i \in \{0, 1\}, \quad i, j = 0, 1, 2, \dots, n$$

(CFLPOT)에서 목적함수는 생산시설 설치비용과 설치된 생산시설로부터 각 노드의 총수요량을 충족시키는 데 소요되는 총비용의 합을 나타낸다. 제약식 (1)은 생산시설 용량에 대한 제약식(capacity constraints)으로써 한 생산시설로부터 충족되어지는 총수요량은 그 생산시설의 용량을 초과할 수 없음을 나타낸다. 제약식 (2)는 수요량의 불가분성에 대한 제약식(indivisible demand constraints)으로써 각 노드의 총수요량은 한 생산시설로부터 전량 충족되어짐을 의미한다. 제약식 (3)은 인접 노드에 대한 제약식(contiguity constraints)으로써 만약 한 노드의 총수요량이 한 생산시설로부터 충족되어진다면 그 노드의 직상위 노드 또한 동일한 생산시설로부터 충족되어짐을 나타내며, 결과적으로 그 노드와 그 생산시설을 연결하는 경로상에 있는 모든 노드의 총수요량이 동일한 생산시설로부터 충족되어짐을 나타낸다.

**3. 나무분할문제**

본 절에서는 CFLPOT가 나무분할문제(tree partitioning problem)

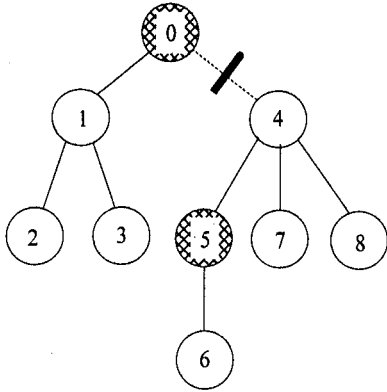


그림 2. 두 개의 부분나무로 분할된 전체 나무.

로 공식화될 수 있음을 보인다. 나무  $T=(V, E)$ 상의 한 노드에 위치한 생산시설로부터 수요량이 충족되어지는 노드들의 집합을  $V'$ 이라고 하고,  $V'$ 상의 각 노드들을 연결하는 호들의 집합을  $E'$ 이라고 하면, 2절에서 제시된 CFLPOT에 대한 두 가지 가정(indivisible demand assumption과 contiguity assumption)에 의해  $T'=(V', E')$ 은 나무  $T$ 의 부분나무(subtree)를 형성하며, 또한 나무  $T$ 가 이러한 부분나무들로 분할(partition)되어짐을 알 수 있다. 예컨대 <그림 1>에서 노드 0, 1, 2, 3에서 요구되는 총수요량은 노드 0에 위치한 생산시설로부터 충족되어지고 노드 4, 5, 6, 7, 8에서 요구되는 총수요량은 노드 5에 위치한 생산시설로부터 충족되어진다고 가정하면, <그림 2>에서 보는 바와 같이 전체 나무가 생산시설이 위치한 노드 0과 5를 중심으로 두 개의 부분나무로 분할되어짐을 알 수 있다.

만약  $T'$ 상에 설치되어지는 어떠한 생산시설도  $T'$ 의 총수요량을 충족시킨다(즉,  $\sum_{j \in T'} d_j \leq \min\{b_i | i \in T'\}$ )고 가정할 때,  $T'$ 상의 한 노드  $i$ 에 위치한 생산시설로부터  $T'$ 의 총수요량을 충족시키는 데 드는 총비용  $c_{T'}^i$ 는 다음과 같이 구해진다.

$$c_{T'}^i = \sum_{j \in T'} c_{ij} + f_i \quad (4)$$

따라서  $T'$ 의 총수요량을 충족시키는 데 드는 최소비용  $c_{T'}$ 과 이와 관련된 생산시설의 위치  $i^*$ 는  $c_{T'} = \min_{i \in T'} c_{T'}^i$ 와  $i^* = \arg \min_{i \in T'} c_{T'}^i$ 로 각각 구해진다.

이제 임의의 부분나무  $\bar{T}=(\bar{V}, \bar{E})$ 에 대하여 생산시설이 노드  $i \in \bar{T}$ 에 위치해 있다고 가정하고 다음과 같은 열 벡터(column vector)들을 정의한다.

$$x_{\bar{T}}^i = (x_{ij}) \in \{0, 1\}^{|\bar{T}|}, \quad x_{\bar{T}} = (x_{\bar{T}}^i) \in \{0, 1\}^{|\bar{T}|}, \\ x = (x_{\bar{T}}) \in \{0, 1\}^{|\mathcal{S}|}, \quad c = (c_{\bar{T}}) \in \{0, 1\}^{|\mathcal{S}|}$$

여기서  $|\bar{T}|$ 는  $\bar{T}$  내에 있는 노드들의 개수를 나타내고,  $|\mathcal{S}|$ 는 나무  $T$ 의 모든 부분나무들의 개수를 나타낸다.

이때  $A$ 를 나무  $T$ 의 node-subtree incidence matrix라고 하면 CFLPOT는 다음과 같은 나무분할문제에 공식화되어진다.

$$(P) \quad \min c^T x \\ \text{s.t. } Ax = 1 \\ x \in \{0, 1\}^{|\mathcal{S}|}$$

문제 (P)는 0-1 정수계획 문제이지만 다행히도  $A$ 의 특성을 이용하면 (P)를 선형계획 문제로 완화시켜 풀어도 여전히 정수해를 갖는다는 사실을 증명할 수 있기 때문에(Chvatal, 1975; Shaw, 1993), 문제 (P)를 푸는 대신 다음의 선형계획 문제 (P')을 풀면 된다.

$$(P') \quad \min c^T x \\ \text{s.t. } Ax = 1 \\ x \in [0, 1]^{|\mathcal{S}|}$$

그러나 나무의 노드 개수가 증가함에 따라  $T$ 의 부분나무의 개수는 지수적으로 증가하기 때문에 문제 (P')은 지수적으로 많은 변수들을 갖는다. 따라서 이와 같은 문제는 일반적으로 Gilmore와 Gomory(1961)에 의해 고안된 열 생성기법으로 풀 수 있으나, 본 문제의 특수한 구조를 이용하여 Shaw(1993)에 의해 개발된 소위 제한된 열 생성기법의 핵심 아이디어를 확장·적용함으로써 좀 더 효율적으로 풀 수 있는데 이에 대한 설명은 5절에서 하기로 한다.

#### 4. 부분 최대감소비용

본 절에서는 제한된 열 생성기법에서 가장 중요하게 이용되는 개념인 문제 (P')에 대한 부분 최대감소비용을 정의하고 이를 효율적으로 구하는 깊이우선 동적계획 알고리즘을 개발한다.

$y_j$ 를 문제 (P')의  $j$ 번째 제약식(즉,  $j$ 번째 노드)에 대한 쌍대변수(dual variable)라고 하면, 변수  $x_{\bar{T}}$ 의 감소비용  $z_{\bar{T}}$ 는  $z_{\bar{T}} = \sum_{j \in \bar{T}} y_j - c_{\bar{T}}$ 로 정의된다. 따라서  $r(\bar{T})$ 를  $\bar{T}$ 의 루트노드라고 할 때(즉,  $r(\bar{T}) = \min\{j | j \in \bar{T}\}$ ), 주어진 노드  $k \in \bar{T}$ 에 대하여 부분 최대감소비용  $z_k^*$ 를  $z_k^* = \max_{r(\bar{T})} r(\bar{T}) = k z_{\bar{T}}$ 로 정의한다. 이때  $z_k^*$ 는 다음과 같이 구해진다.

$$z_k^* = \max_{r(\bar{T})=k} z_{\bar{T}} \\ = \max_{r(\bar{T})=k} \left( \sum_{j \in \bar{T}} y_j - c_{\bar{T}} \right) \\ = \max_{r(\bar{T})=k} \left( \sum_{j \in \bar{T}} y_j - \min_{i \in \bar{T}} c_{\bar{T}}^i \right) \\ = \max_{r(\bar{T})=k} \max_{i \in \bar{T}} \left( \sum_{j \in \bar{T}} y_j - c_{\bar{T}}^i \right) \\ = \max_{i \in T(k)} \max_{i \in \bar{T}, r(\bar{T})=k} \left( \sum_{j \in \bar{T}} y_j - c_{\bar{T}}^i \right) \\ = \max_{i \in T(k)} z_k^i$$

여기서  $T(k) = (V_k, E_k)$ 는  $r(T(k)) = k$ 인  $T$ 의 완전부분 나무(complete subtree)로서  $V_k = \{j \in V \mid k \in \gamma[0, j]\}$ 로 정의되며, 또한  $z_k^i$ 는 다음과 같이 정의된다.

$$z_k^i = \max_{i \in T, r(T) = k} \left( \sum_{j \in T} y_j - c_{ij}^i \right) \quad (5)$$

따라서 주어진 노드  $k \in T$ 와 모든 노드  $i \in T(k)$ 에 대하여 식 (5)와 같이 정의된  $z_k^i$ 의 값만 구하면 부분 최대감소비용  $z_k^*$ 는 쉽게 구해짐을 알 수 있다.

이제  $z_k^i$ 의 값을 구하기 위하여 고정된 노드  $i \in T(k)$ 에 위치한 생산시설을 기준으로 모든 노드  $j \in T(k)$ 에 대하여 결정 변수  $\xi_j$ 를 다음과 같이 정의한다.

$$\xi_j = \begin{cases} 1, & \text{만약 노드 } i \text{에 있는 생산시설이 노드 } j \text{의 총수요를 충족시키면,} \\ 0, & \text{그렇지 않으면.} \end{cases}$$

이때  $\bar{c}_{ij} = y_j - c_{ij}$ 라고 하면  $z_k^i$ 는 식 (4)와 (5)에 의해 다음과 같은 제한용량이 있는 나무배낭문제 ( $\bar{P}(i, k, b_i)$ )의 최적값  $\Psi_{T(k)}(i, k, b_i)$ 을 이용함으로써  $z_k^i = \Psi_{T(k)}(i, k, b_i) - f_i$ 로 쉽게 구할 수 있다.

$$\begin{aligned} \Psi_{T(k)}(i, k, b_i) &= \max \sum_{j \in T(k)} \bar{c}_{ij} \xi_j \\ \text{s.t. } \xi_j &\geq \xi_i, \quad j \in T(k) \setminus \{i\} \\ (\bar{P}(i, k, b_i)) \quad &\sum_{j \in T(k)} d_j \xi_j \leq b_i \\ &\xi_k = 1 \\ &\xi_j \in \{0, 1\} \end{aligned}$$

여기서 우리는 편의상  $\sum_{j \in T(k)} d_j > b_i$ 라고 가정한다. 왜냐하면,  $\sum_{j \in T(k)} d_j \leq b_i$ 인 경우에는 문제 ( $\bar{P}(i, k, b_i)$ )에서 제약식  $\sum_{j \in T(k)} d_j \xi_j \leq b_i$ 를 제거할 수 있으므로 문제 ( $\bar{P}(i, k, b_i)$ )가 제한용량이 없는 나무배낭문제가 되어  $O(|T(k)|) \approx O(n)$ 시간으로 쉽게 풀 수 있기 때문이다(Cho, 1998).

문제 ( $\bar{P}(i, k, b_i)$ )를 풀기 전에, 우선 이의 특별한 문제인 ( $\bar{P}(i, i, b_i)$ ), 즉 루트노드에 생산시설이 위치한 경우의 제한용량이 있는 나무배낭문제를 푸는 깊이우선 동적계획 알고리즘들을 먼저 제시한다. 사실, 문제 ( $\bar{P}(i, k, b_i)$ )는 문제 ( $\bar{P}(i, i, b_i)$ )에 대한 깊이우선 동적계획 알고리즘들을 확장·적용함으로써 쉽게 풀 수 있는데, 이에 대한 자세한 설명은 뒤에서 하기로 한다.

문제 ( $\bar{P}(i, i, b_i)$ )에 대한 깊이우선 동적계획 알고리즘들의 기본 아이디어는 루트노드에서 시작하여 깊이우선 탐색순서에 따라 다음에서 정의되는 '전방이동(forward move)'을 적용

하고 전방이동이 불가능할 경우(즉, 잎노드(leaf node)에 도달하였거나 더 이상 탐색할 후행노드(successor)가 없는 경우)에는 '후방이동(backward move)'을 적용하는 것이다. 이는 루트노드  $i$ 의 마지막 직하위 후행노드(immediate successor)  $w^*$ (즉,  $w^* = \max\{j \mid p_j = i\}$ )에서  $i$ 로 후방이동이 이루어질 때까지 수행된다.

집합  $L$ 을 깊이우선 탐색순서에 의해 탐색된 노드들의 집합이라고 하면 전방이동과 후방이동은 다음과 같이 정의된다.

1. (초기화)  $\Psi_{(i)}(i, i, h) = \begin{cases} \bar{c}_{ii} & \text{if } d_i \leq h \leq b_i; \\ -\infty & \text{otherwise} \end{cases}$ ;
2. ( $p_v$ 에서  $v$ 로 전방이동)  $L \leftarrow L \cup \{v\}$ ;

$$\Psi_L(i, v, h) = \begin{cases} \Psi_{L \setminus \{v\}}(i, p_v, h - d_v) + \bar{c}_{iv} & \text{if } h \geq \sum_{j \in T(i, v)} d_j; \\ -\infty & \text{otherwise} \end{cases}$$

3. ( $v$ 에서  $p_v$ 로 후방이동) 모든  $h = 0, 1, 2, \dots, b_i$ 에 대하여  $\Psi_L(i, p_v, h) = \max\{\Psi_{L \setminus T(v)}(i, p_v, h), \Psi_L(i, v, h)\}$ ;

또한 문제 ( $\bar{P}(i, i, b_i)$ )에 대한 최적해를 구하기 위해 다음과 같은 값  $I_L(i, v, h)$ 를 정의한다.

1. (초기화)  $I_{(i)}(i, i, h) = \begin{cases} 1 & \text{if } d_i \leq h \leq b_i; \\ 0 & \text{otherwise} \end{cases}$ ;
2. ( $v$ 에서  $p_v$ 로의 후방이동)  $I_L(i, v, h) = \begin{cases} 1 & \text{if } \Psi_{L \setminus T(v)}(i, p_v, h) < \Psi_L(i, v, h); \\ 0 & \text{otherwise} \end{cases}$ ;

이제 문제 ( $\bar{P}(i, i, b_i)$ )를 푸는 깊이우선 동적계획 알고리즘들을 구체적으로 제시하기로 한다.

- 단계 0. (초기화)  $u \leftarrow i$ ;  $temp \leftarrow d_i$ ;

$$\Psi(i, i, h) = \begin{cases} \bar{c}_{ii} & \text{if } d_i \leq h \leq b_i; \\ -\infty & \text{otherwise} \end{cases}$$

$$I(i, i, h) = \begin{cases} 1 & \text{if } d_i \leq h \leq b_i; \\ 0 & \text{otherwise} \end{cases}$$

- 단계 1. 만약 노드  $u$ 가 잎노드가 아니면  $u \leftarrow u + 1$ 로 하고 단계 2로 가고 그렇지 않으면  $v \leftarrow u$ 로 하고 단계 3으로 간다;

- 단계 2. (전방이동)

- i)  $temp \leftarrow temp + d_u$ ;

- ii) 모든  $h = 0, 1, \dots, temp - 1$ 에 대하여  $\Psi(i, u, h) \leftarrow -\infty$ ;

- iii) 모든  $h = temp, temp + 1, \dots, b_i$ 에 대하여  $\Psi(i, u, h) \leftarrow \Psi(i, p_u, h - d_u) + \bar{c}_{iu}$ ;

iv) 단계 1로 간다;

단계 3. (후방이동)

- i)  $temp \leftarrow temp - d_v$ ;
- ii) 모든  $h=0, 1, \dots, b_i$ 에 대하여 만약  $\Psi(i, p_v, h) < \Psi(i, v, h)$ 이면  $\Psi(i, p_v, h) \leftarrow \Psi(i, v, h)$ ;  $I(i, v, h) \leftarrow 1$ ; 그렇지 않으면  $I(i, v, h) \leftarrow 0$ ;

iii) 단계 4로 간다;

- 단계 4. i) 만약  $p_v = p_{u+1}$ 이면  $u \leftarrow u + 1$ 로 두고 단계 2로 간다;
- ii) 만약  $p_v \neq p_{u+1}$ 이고  $p_v \neq i$ 이면  $v \leftarrow p_v$ 로 두고 단계 3으로 간다;
- iii) 만약  $p_v \neq p_{u+1}$ 이고  $p_v = i$ 이면 수행을 종료한다.

위 알고리즘에서 우리는 앞에서 소개한  $\Psi_L(i, v, h)$ ,  $I_L(i, v, h)$ 를 사용하는 대신에 침자  $L$ 이 제거된  $\Psi(i, v, h)$ ,  $I(i, v, h)$ 를 각각 사용하였다. 이는 깊이우선 탐색순서에 따라 알고리즘이 수행되면서  $\Psi(i, v, h)$ ,  $I(i, v, h)$ 값들이 새롭게 계산된 값들로 자동적으로 대체되므로 굳이 깊이우선 탐색순서에 의해 탐색된 노드들의 집합  $L$ 을 기억해 둘 필요가 없기 때문이다. 또한 단계 4에서  $u = n$ 이면  $p_{u+1}$ 은 정의되지 않으나, 편의상  $p_{n+1} = -1$ 로 정의한다.

이제 위에서 제시된 알고리즘의 시간 복잡도(time complexity)를 계산해 보기로 한다. 위 알고리즘의 주된 연산들(operations)은 전방이동과 후방이동으로써 각각 나무  $T(i)$ 상의 호의 개수회(즉,  $|T(i)| - 1$ )만큼 수행되어지며 또한 각 이동은  $O(b_i)$  시간이 소요되므로, 결과적으로 위 알고리즘은  $O(|T(i)|b_i)$  시간이 소요됨을 알 수 있다.

문제  $(\bar{P}(i, k, b_i))$ 를 푸는 깊이우선 동적계획 알고리즘은 문제  $(\bar{P}(i, i, b_i))$ 를 푸는 깊이우선 동적계획 알고리즘으로부터 쉽게 도출될 수 있음을 보이기 위해 먼저  $(\bar{P}(i, p_i, b_i))$ 를 푸는 깊이우선 동적계획 알고리즘을 도출한다. 나무  $T(p_i)$ 는 <그림 3>에서 보는 바와 같이  $T(p_i) = T(i) \cup (T(p_i) \setminus T(i))$ 로 변형(deformation)될 수 있으므로 문제  $(\bar{P}(i, p_i, b_i))$

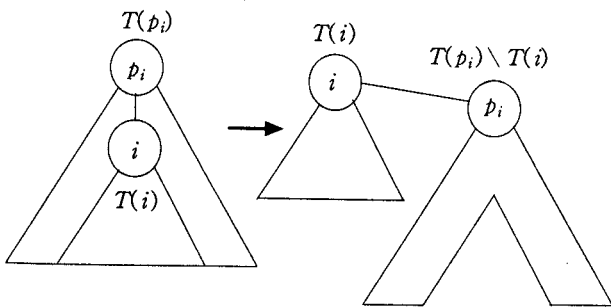


그림 3.  $T(p_i) = T(i) \cup (T(p_i) \setminus T(i))$ .

를 푸는 깊이우선 동적계획 알고리즘은 문제  $(\bar{P}(i, i, b_i))$ 를 푸는 깊이우선 동적계획 알고리즘으로부터 얻어진 결과들(즉, 알고리즘 종료 후 얻어진  $\Psi(i, v, h)$ ,  $I(i, v, h)$ ,  $temp$  값들)을 이용하여 먼저  $L = T(i)$ 로 놓고 노드  $i$ 에서 노드  $p_i$ 로 전방이동을 수행한 후(이때  $L = T(i) \cup \{p_i\}$ 로 변함), 나무  $T(p_i) \setminus T(i)$ 상에 연속적으로 위에서 제시된 깊이우선 동적계획 알고리즘을 확장·적용함으로써 도출되며, 이때 도출된 알고리즘의 시간복잡도는  $O((|T(i)| + |T(p_i) \setminus T(i)|)b_i) = O(|T(p_i)|b_i)$ 시간이다.

이와 같은 방법을 연속적으로 적용함으로써 문제  $(\bar{P}(i, k, b_i))$ 를 푸는 깊이우선 동적계획 알고리즘은  $T(k) = T(p_k) \cup (T(k) \setminus T(p_k))$ 임을 이용하여 문제  $(\bar{P}(i, p_k, b_i))$ 를 푸는 깊이우선 동적계획 알고리즘으로부터 쉽게 도출됨을 알 수 있으며, 시간 복잡도는  $O((|T(p_k)| + |T(k) \setminus T(p_k)|)b_i) = O(|T(k)|b_i) \approx O(nb)$ 시간이 되므로, 결과적으로  $z_k^i = \Psi_{T(k)}(i, k, b_i) - f_i$ 로 정의되는 부분 최대감소비용은  $O(nb)$ 시간으로 구해질 수 있음을 알 수 있다.

### 5. 제한된 열 생성기법

본 절에서는 Shaw(1993)에 의해 개발된 소위 제한된 열 생성기법의 핵심 아이디어를 확장·적용함으로써 문제  $(P)$ 을  $O(n^2b)$ 시간으로 풀 수 있음을 보인다.

$(n+1) \times (n+1)$  행렬  $B$ 를  $(P)$ 의 제약식에 대한 기저(basis)라고 하고  $I$ 를  $(n+1) \times (n+1)$  항등행렬이라고 하자. 또한  $\bar{T}_k^* = \arg \max_{r(T)=k} z_{\bar{T}}$  (즉,  $z_k^* = z_{\bar{T}_k^*}$ )라고 하고 벡터  $1_{\bar{T}_k^*} = (\omega_i) \in \{0, 1\}^{n+1}$ 의 성분  $\omega_i$ 를 다음과 같이 정의하자.

$$\omega_i = \begin{cases} 1 & \text{if } i \in \bar{T}_k^* \\ 0 & \text{otherwise.} \end{cases}$$

이때  $(P)$ 을 푸는 제한된 열 생성기법의 기본 아이디어는 처음에  $B = I$ ,  $y_k = c_{(k)} = c_{ii} + f_i$ 로 두고 시작하여 역깊이우선 탐색순서(reversed depth first search order)인  $k = n, n-1, \dots, 0$ 를 따라 진행되는 심플렉스 해법(simplex method)의 매 단계마다 부분 최대감소비용을 계산하여 이 값이 양수이면(즉,  $z_k^* > 0$ 이면)  $B$ 의  $k$ 번째 열벡터를  $1_{\bar{T}_k^*}$ 로 바꾸고  $y_k, z_k^*$ 를 각각 갱신(update)하는 과정을 반복 수행하는 것이다. 이는 처음에 모든 노드  $i=0, 1, 2, \dots, n$ 에 각각 생산시설을 설치함으로써 각 노드의 총수요량을 그 노드에 설치된 생산시설로부터 충족시키는 것에서 시작하여 점차 완전부분나무  $T(k)$ 의 루트노드  $k$ 를  $k = n, n-1, \dots, 0$ 순으로 옮겨 가면

서  $T(k)$ 상에서의 최적의 생산시설의 위치 및 개수를 효율적으로 조정해 가는 것을 의미한다.

이제 (P)을 푸는 제한된 열 생성기법의 상세한 절차를 소개하면 다음과 같다.

단계 0. (초기화)

$B \leftarrow I$ ; 모든  $j = 0, 1, 2, \dots, n$ 에 대하여  $y_j \leftarrow c_{(j)}$ ;  
 $k \leftarrow n$ ;

단계 1. 다음을 수행한다;

- i)  $z_k^*$ 를 계산하고  $\overline{T}_k^*$ 를 찾는다;
- ii) 만약  $z_k^* > 0$ 이면 다음을 수행한다;

- ① 행렬  $B$ 의  $k$ 번째 열을  $1/\overline{T}_k^*$ 로 대체(replace)한다;
- ②  $y_k \leftarrow y_k - z_k^*$ ;
- ③ 모든  $i \in T(k)$ 와 모든  $h = 0, 1, \dots, b$ 에 대하여

$$\Psi(i, k, h) \leftarrow \Psi(i, k, h) - z_k^*;$$

- ④ 모든  $i \in T(p_0^b)$ 에 대하여  $\overline{c}_{ik} \leftarrow \overline{c}_{ik} - z_k^*$ ;

iii)  $k \leftarrow k - 1$ ;

단계 2. 만약  $k \geq 0$ 이면 단계 1로 가고 그렇지 않으면

$$\Psi^* \leftarrow \sum_{k=0}^n y_k;$$

단계 3.  $B$ 를 탐색하여 최적해를 구한다;

위의 단계 2에서  $\Psi^*$ 는 문제 (P)의 최적값을 나타내며, 이는 쌍대변수의 합으로 쉽게 구해짐을 알 수 있다. 또한 단계 3에서 최적해를 구하는 기본 아이디어는 다음과 같다. 먼저  $B$ 의 첫 번째 열을 탐색하여 1을 갖는 노드들을 찾아낸다. 이를 이용하여 부분나무를 구성하면 당연히  $\overline{T}_0^*$ 가 되는데 이는  $\overline{T}_0^*$  내에 있는 한 노드에 위치한 생산시설(이 위치는 쉽게 찾을 수 있음)이  $\overline{T}_0^*$ 에 있는 각 노드의 총수요를 충족시키는 것이 최적임을 의미한다. 다음으로  $\overline{T}_0^*$ 의 앞노드들을  $l_1, l_2, \dots, l_t$ 라고 할 때, 각  $j = 1, 2, \dots, t$ 에 대하여  $l_j$ 의 직후행노드들의 집합  $S_j = \{u \mid p_u = l_j\}$ 에 있는 각 원소  $u$ 에 대응되는  $B$ 의  $u$ 번째 열을 탐색하여 1을 갖는 노드들을 찾아낸다. 이와 같은 방법을 계속 적용하면 <그림 4>에서 보는 바와 같이 나무  $T$ 가 부분나무들로 분할(partition)되어지며 궁극적으로 문제 (P)에 대한 최적해가 구해진다.

이제 위에서 소개된 제한된 열 생성기법과 4절에서 제시한 깊이우선 동적계획 알고리즘을 이용하여 문제 (P)을  $O(n^2b)$  시간으로 풀 수 있음을 보인다. 이를 위하여 먼저 각 단계별로 소요되는 시간을 계산해 보기로 한다. 단계 0는  $B$ 가  $(n+1) \times (n+1)$  행렬이므로  $B \leftarrow I$ 를 위해서는  $O(n^2)$  시간이 소요되며 단계 1은 i)과 ii)의 ③을 수행하는 데 소요되는 시간에 의해 결정되어진다. 실제로 i)에서는  $z_k^* = \max_{i \in T(k)} z_k^i$ 이고

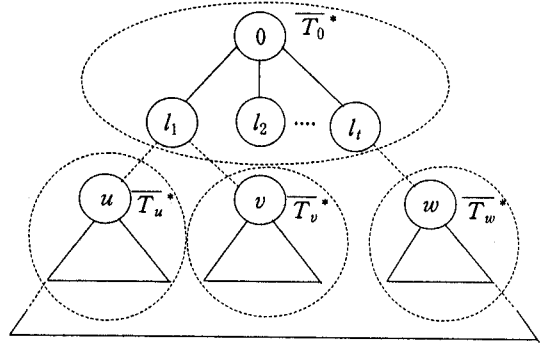


그림 4. 제한된 열 생성기법의 최적해 도출 방법과 나무  $T$ 의 분할.

$z_k^i = \Psi_{T(k)}(i, k, b) - f_i$ 이므로 4절에서 보았듯이  $\Psi_{T(k)}(i, p_k^i, b)$ 를 이용하여  $\Psi_{T(k)}(i, k, b)$ 를 계산하는데  $O(|T(k) \setminus T(p_k^i)|b)$  시간이 소요된다. 따라서  $z_k^*$ 를 계산하는데  $O(|T(k)|b) + O(\sum_{i \in T(k) \setminus \{k\}} |T(k) \setminus T(p_k^i)|b)$  시간이 소요되는데, 단계 1은 단계 2에 의해 역깊이우선 탐색순서인  $k = n, n-1, \dots, 0$ 를 따라  $n+1$ 번 수행되어지며, 또한  $i \in T(k) \setminus \{k\} \Leftrightarrow k \in \gamma(i, 0)$ 가 성립하므로 <그림 5>에서 보는 바와 같이 단계 1의 i)에 소요되는 총수행시간은 다음과 같이 구할 수 있다.

$$\begin{aligned} & O\left(\sum_{k=0}^n (|T(k)|b + \sum_{i \in T(k) \setminus \{k\}} |T(k) \setminus T(p_k^i)|b)\right) \\ &= O\left(\sum_{i=0}^n (|T(i)|b + \sum_{k \in \gamma(i, 0)} |T(k) \setminus T(p_k^i)|b)\right) \\ &= O\left(\sum_{i=0}^n (|T(i)| + |T(0)| - |T(i)|)b\right) \\ &= O(n^2b) \end{aligned}$$

단계 1의 ii)의 ③도 i)에서의 마찬가지로 단계 2에 의해 역깊이우선 탐색순서인  $k = n, n-1, \dots, 0$ 를 따라  $n+1$ 번 수행되므로 총수행시간은  $O(n^2b)$ 임을 쉽게 알 수 있다. 또한 단

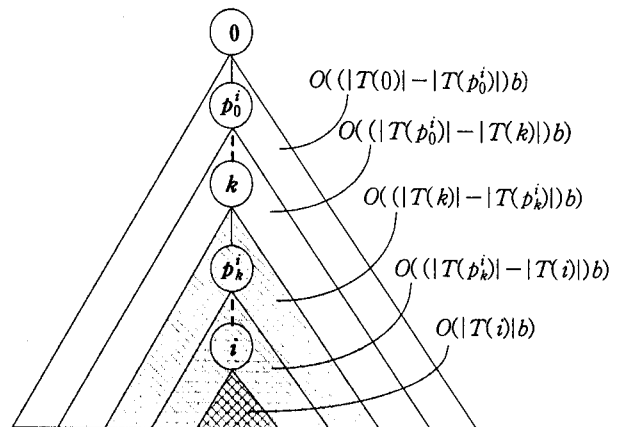


그림 5. 부분최대감소비용 계산시 고정된 노드  $i$ 와 관련된 총수행시간

계 2는 상수 시간(constant time)이 소요되며 단계 3은 나무 탐색 알고리즘(tree search algorithm)에 의해  $O(n)$ 시간이 소요된다.

결과적으로 지금까지 구한 각 단계별 수행 시간을 종합해 보면 제한된 열 생성기법으로 문제 ( $P$ )을 푸는 데 소요되는 총 수행시간은  $O(n^2b)$ 임을 알 수 있다.

### 6. 실험결과

본 절에서는 본 연구에서 제시된 제한된 열 생성기법과 깊이 우선 동적계획 알고리즘을 C 언어로 코딩하여 여러 가지 유형의 나무구조를 갖는 네트워크 상에서의 제한 용량이 있는 입지설정문제(CFLPOT)에 대한 테스트를 실시한다. 사실 본 연구에서 제시된 알고리즘의 효율성을 검증하기 위해서는 문헌상에 보고된 테스트 결과와의 비교·검토가 수행되어야 하지만 본 연구에서 제시된 CFLPOT 모형과 동일한 모형에 대한 연구결과의 부재로 인하여 본 연구에서는 다음과 같은 방법으로 여러 가지 유형의 문제들을 임의로 생성하여 테스트를 실시하고자 한다.

먼저 나무  $T$ 의 구조를 임의로 결정한다. 이를 위해 나무  $T$ 의 총노드 수  $n$ 을 고정한 후 루트노드로부터 시작하여 직하 위노드 수를 구간  $[0, \log_2 n]$ 에서 임의로 결정하고 이를 넓이우선탐색(breadth first search : BFS)순서로 나열한 후, 다시 생성된 각 노드의 직하위노드 수를 구간  $[0, \log_2 n]$ 에서 임의로 결정하고 이를 BFS 순서로 나열하는 방법을 반복적으로 수행하는데 이는 생성된 총노드 수가  $n$ 이 될 때까지 계속 된다.

본 실험에서는 편의상  $b = \max \{b_i | i \in T\}$ 의 값은 500과 1,000을 갖는 두 가지 경우로 한정하고 각  $b_i$ 는 구간  $[\frac{b}{2}, b]$ 에서 임의로 결정하며,  $b = 500$ 인 경우에는  $d_i$ 를 구간  $[1, 1.50]$ 에서,  $b = 1,000$ 인 경우에는  $d_i$ 를 구간  $[1, 100]$ 에서 임의로 결정한다. 또한 각  $c_i$ 는 구간  $[1, 50]$ 에서 임의로 결정하며,  $f_i$ 를 두 개의 다른 구간  $[1, 1000]$ ,  $[9000, 10000]$ 에서 임의로 결정하고 또한 각각의 경우에 대하여 i)  $t_v \in [1, 50]$ ,  $D_v \in [1, 100]$  ii)  $t_v \in [1, 10]$ ,  $D_v \in [1, 20]$ 에서  $t_v, D_v$ 를 각각 임의로 결정함으로써 생산시설 설치비용과 수송비용의 크기에 대한 비율의 변화가 어떻게 생산시설 설치 개수에 영향을 주는지를 살펴보고자 한다.

본 실험에서는 문제 유형을 총노드 수에 따라 6가지 ( $n = 10, 20, 40, 70, 100, 150$ )로 구분하고 이를 다시  $b = 500, 1000$ 의 두 가지 값에 따라 구분한 후 각 유형별로 20문제를 임의로 생성하여 SUN Ultra Enterprise 2 환경에서 테스트를 실시한다. <표 1>, <표 2>, <표 3>, <표 4>는 테스트 결과로부터 도출된 각 유형별 CPU 시간과 생산시설 설치 개수들의 평균값(average value), 가장 나쁜 값(worst value), 가장 좋은 값(best value) 등을

표 1. CFLPOT의 각 유형별 테스트 결과

( $f_i \in [1, 1000]$ ,  $t_v \in [1, 50]$ ,  $D_v \in [1, 100]$ 인 경우)

n	b	CPU 시간(단위: 초)			생산시설 설치 개수		
		worst	average	best	worst	average	best
10	500	0.30	0.20	0.15	10	9	8
	1,000	0.55	0.34	0.27	10	9	8
20	500	1.36	0.80	0.35	20	19	18
	1,000	2.32	1.41	0.61	20	19	17
40	500	3.18	1.69	0.93	40	39	37
	1,000	5.79	3.19	1.53	39	38	33
70	500	9.31	3.48	2.19	70	68	65
	1,000	17.73	6.36	4.52	69	67	63
100	500	31.11	10.87	6.90	99	98	93
	1,000	60.74	21.01	11.34	98	95	92
150	500	70.30	31.63	12.62	150	146	138
	1,000	137.45	77.84	17.73	146	138	135

표 2. CFLPOT의 각 유형별 테스트 결과

( $f_i \in [9000, 10000]$ ,  $t_v \in [1, 50]$ ,  $D_v \in [1, 100]$ 인 경우)

n	b	CPU 시간(단위: 초)			생산시설 설치 개수		
		worst	average	best	worst	average	best
10	500	0.35	0.27	0.15	10	8	7
	1,000	0.60	0.41	0.25	8	6	3
20	500	1.37	0.86	0.42	19	15	12
	1,000	2.32	1.57	0.80	18	14	11
40	500	3.06	1.74	0.96	34	28	26
	1,000	5.85	3.36	1.77	33	23	20
70	500	9.39	5.67	2.52	62	53	42
	1,000	18.06	10.73	4.70	53	46	38
100	500	31.61	14.22	7.59	82	81	71
	1,000	60.70	28.98	14.05	75	71	59
150	500	70.14	25.32	15.65	126	114	107
	1,000	136.80	76.77	14.49	109	95	94

나타내 주고 있다. 각 표에서 볼 수 있듯이 CPU 시간은  $f_i, t_v, D_v$  값들의 변화에 대해 민감하지 않음을 알 수 있으며,  $n$ 에 대해서는 근사적으로 제곱의 비율(quadratic rate)로,  $b$ 에 대해서는 근사적으로 선형의 비율(linear rate)로 변화함을 알 수 있다. 반면에 생산시설 설치 개수는  $f_i, t_v, D_v$  값들의 변화에 대해 매우 민감함을 알 수 있다. 예컨대, <표 1>과 <표 2>, <표 3>과 <표 4>를 각각 살펴보면, 생산시설 설치비용의 증가율이 수송비용의 증가율보다 상대적으로 커지면 생산시설 설치 개수가 감소함을 알 수 있으며, 특히 <표 1>, <표 2>, <표 3>보다 <표 4>의 경우에 생산시설 설치 개수가 현저히 감소함을 알



표 3. CFLPOT의 각 유형별 테스트 결과

( $f_i \in [1, 1000]$ ,  $t_v \in [1, 10]$ ,  $D_v \in [1, 20]$ 인 경우)

n	b	CPU 시간(단위 : 초)			생산시설 설치 개수		
		worst	average	best	worst	average	best
10	500	0.34	0.22	0.16	10	7	5
	1,000	0.61	0.39	0.25	8	6	5
20	500	1.35	0.89	0.44	18	13	8
	1,000	0.27	1.53	0.72	18	12	9
40	500	3.12	1.77	0.93	31	28	22
	1,000	5.85	3.29	1.59	30	24	19
70	500	9.38	3.66	2.37	56	44	39
	1,000	18.03	6.72	4.86	50	40	35
100	500	31.26	11.28	7.20	76	68	62
	1,000	61.30	21.95	12.31	68	56	50
150	500	70.00	31.47	12.95	112	100	89
	1,000	138.84	78.42	18.21	104	91	80

표 4. CFLPOT의 각 유형별 테스트 결과

( $f_i \in [9000, 10000]$ ,  $t_v \in [1, 10]$ ,  $D_v \in [1, 20]$ 인 경우)

n	b	CPU 시간(단위 : 초)			생산시설 설치 개수		
		worst	average	best	worst	average	best
10	500	0.41	0.24	0.14	6	3	1
	1,000	0.72	0.51	0.30	2	2	1
20	500	1.49	1.00	0.53	7	5	2
	1,000	2.55	1.78	0.89	4	3	1
40	500	3.42	1.95	1.04	11	7	5
	1,000	6.24	3.62	2.01	6	4	3
70	500	10.23	6.18	2.94	17	14	8
	1,000	18.96	11.37	5.10	11	8	5
100	500	34.03	15.57	8.32	23	20	15
	1,000	63.35	30.92	15.18	15	12	8
150	500	75.59	27.28	17.15	37	25	17
	1,000	144.68	81.29	15.76	23	19	14

수 있다. 이는 <표 1>, <표 2>, <표 3>에서는 임의로 생성된 문제들의 생산시설 설치비용이 수송비용보다 상대적으로 작게 생성됨으로 인하여 가능하면 많은 곳에 생산시설을 설치하는 것이 최적해가 될 수 있는 가능성이 높은 반면에 <표 4>에서는 반대로 수송비용이 생산시설 설치비용보다 작게 생성될 가능성이 높아짐으로 인해 가능하면 생산시설을 적게 설치하여 설치된 시설들로부터 수송을 통해 각 노드에서 발생하는 수요를 충족시키는 것이 최적해가 될 가능성이 높다는 것을 알 수 있다.

### 7. 결론

본 연구에서는 먼저 수요 비분리 가정과 인접성 가정하에서의 나무구조를 갖는 네트워크상에서의 제한용량이 있는 입지설 정문제(CFLPOT)에 대한 일반적인 정수계획 모형을 소개하였고, 다음으로 CFLPOT를 지수적으로 많은 변수를 갖는 나무분 할문제 (P)로 공식화하였다. 또한 문제 (P)의 선형계획 완화 문제 (P')이 정수해를 갖는다는 사실과 지수적으로 많은 변수 를 갖는다는 사실을 이용함으로써 문제 (P)는 Shaw(1993)에 의해 개발된 소위 제한된 열 생성기법의 핵심 아이디어를 확장· 적용하여  $O(n^2b)$ 시간으로 풀 수 있음을 보였다. 아울러 제한된 열 생성기법에서 핵심적인 역할을 하는 부분 최대감소 비용은 제한용량이 있는 나무배낭문제에 대한 최적값으로부터 얻어질 수 있음을 보였고, 이는 본 연구에서 개발된 깊이우 선 동적계획 알고리즘에 의해  $O(nb)$ 시간으로 구할 수 있음 을 보였다. 마지막으로 본 연구에서 제시한 알고리즘을 C 언 어로 코딩하여 여러 가지 유형의 문제를 임의로 생성하여 테 스트를 실시하였다.

본 연구에서 개발된 알고리즘을 이용하면 나무구조를 갖는 네트워크 상에서의 제한용량이 있는 입지설 정문제에 대한 정확하고 빠른 최적해의 도출이 가능하기 때문에, 경영자의 잘못된 입지의사결정으로 인한 막대한 비용과 시간의 낭비를 사전에 방지함으로써 기업의 경쟁력 향상에 큰 도움을 줄 수 있을 것으로 기대된다.

추후에는 본 연구에서 제시된 아이디어를 확장함으로써 나 무구조를 갖는 네트워크 상이 아닌 일반 네트워크(general networks)상에서의 제한용량이 있는 입지설 정문제를 푸는 효율적인 알고리즘을 개발하는 연구가 수행되어야 할 것으로 사료된다.

### 참고문헌

Averbakh, I. and Berman, O. (1998), Location Problems with Grouped Structure of Demand: Complexity and Algorithms, *Networks*, 31, 81-92.

Barcelo, J. and Casanovas, J. (1984), A Heuristic Lagrangean Algorithm for the Capacitated Plant Location Problem, *European Journal of Operational Research*, 15, 212-226.

Cho, G. (1998), An Efficient Algorithm for an Uncapacitated Facility Location Problem on a Tree Structured Network, *DAEHAN Journal of Business*, 18, 97-109.

Chvatal, V. (1975), On Certain Polytopes Associated with Graphs, *Journals of Combinatorial Theory B*, 13, 138-154.

Gilmore, P. C. and Gomory, R. E. (1961), A Linear Programming Approach to the Cutting Stock Problem, *Operations Research*, 9, 849-859.

Holmberg, K., Rönnqvist, M. and Yuan, D. (1999), An Exact Algorithm for the Capacitated Facility Location Problems with Single Sourcing, *European Journal of Operational Research*, 113, 544-559.

- Klincewicz, J. and Luss, H. (1986), A Lagrangian Relaxation Heuristic for Capacitated Facility Location with Single-source Constraints, *Journal of Operational Research Society*, **37**, 495-500.
- Kolen, A. (1983), Solving Covering Problems and the Uncapacitated Plant Location Problem on Trees, *European Journal of Operational Research*, **12**, 266-278.
- Megiddo, N., Zemel, E. and Hakimi, S. L. (1983), The Maximum Coverage Location Problem, *SIAM Journal on Algebraic and Discrete Methods*, **4**, 253-261.
- Mirchandani, P. B. and Francis, R. L. (1990), *Discrete Location Theory*, John Wiley and Sons, Inc. New York, U.S.A.
- Pirkul, H. (1987), Efficient Algorithm for the Capacitated Concentrator Location Problem, *Computers and Operations Research*, **14**, 197-208.
- Shaw, D. X. (1993), Reformulation and Column Generation for Several Telecommunication Network Design Problems, Technical Report, School of Industrial Engineering, Purdue University, West Lafayette, IN47907, U.S.A.
- Sridharan, R. (1993), A Lagrangian Heuristic for the Capacitated Plant Location Problem with Single Source Constraints, *European Journal of Operational Research*, **66**, 305-312.