〈研究論文〉

Journal of the Korean Institute of Industrial Engineers Vol. 12, No. 1, June 1986

倉庫立地選定問題의 最適解法에 관한 硏究
(A Study on the Optimal Warehouse Location Problem by Using the Branch & Bound Algorithm)

李 得 雨*
李 相 鎔*

Abstract

This paper deals with the problem of the optimal location of warehouses in the two stage distribution system, i.e., the distribution system where the product is transported from plants to customer areas via warehouses.

The Problem is formulated with a zero-one mixed integer programming and an efficient branch and bound algorithm is then used to solve the problem.

In order to obtain the solution of this problem, this paper shows the procedure of conversion of two stage distribution system into one stage distribution system.

An improved method of solving the linear programming at the nodes and branching decision rule is also showed by this study.

1. 序 論

工場과 倉庫의 立地를 選定하는 問題는 여러가지 因子와 관련되어 生產原價에 影響을 주기 때문에 立地選定이 잘못 되어 立地變更이나 設備移轉을 하게 될 때 엄청난 費用과 損失이 따르게 마련이므로 立地選定은 企業의 全體 시스템의 總流通費를 最小化 시키기 위한 長期的인 經營意思決定問題에 속한다.

現在까지의 대부분의 硏究는 주로 1段階 流通構造를 取扱하여 왔다.[1],[5],[6],[8],[11],[12]

그러나 現實的으로 實際의 流通構造는 2段階의 形態(工場으로부터 倉庫를 經由해서 消費地에 製品이 割當)를 많이 볼 수 있다.

이와 같이 2段階 流通시스템에서 倉庫의 立地는 輸送費用(transportation cost)을 줄일 수 있으나 固定費(fixed cost) 즉 開設費用과 運營費用, 上·下車費用, 取扱損失費用, 維持費用을 發生시킨다.

따라서 效果的인 流通시스템을 爲한 最適倉庫의 數와 立地의 決定은 倉庫의 數가 늘어남에 따라 증가하는 固定費用과의 相互關係에서 流通시스템의 總費用을 最小化 시키는 合理的인 方法이 要求된다.

*建国大学校 工科大学 産業工学科

Balinski[12]는 模型의 設定에 融通性을 줄 수 있으나 計算上 非効率的인 整数計劃法(Integer Programming)으로 解를 구하고 있다.

Marks[2]는 2段階의 模型을 네트워크 技法을 利用하여 解法을 제시하고 있으며, Geoffrion & Graves[4]는 Benders 分割法(Benders Decomposition)을 적용하여 解를 구하고 있는데 이는 특히 多品種製品의 경우에 유용하다.

이 밖에도 工場과 倉庫의 立地選定을 同時에 決定하는 Kauman & Eede[9]의 硏究가 있다.

本 硏究에서는 2段階 流通시스템에서 倉庫의 立地候補地가 미리 정해져 있는 倉庫立地選定(Warehouse Location)問題를 混合二進整数計劃法(0-1 Mixed Integer Programming)으로 정식화하고 이의 解를 効果的으로 計算하기 위한 簡略化된 分岐限界法[13](Branch and Bound Algorithm)을 제시하고자 한다.

2. 數學모델의 定立

2-1 模型의 假定

① 製品은 同質의 製品組合의 形態를 取하는 單一品目이다.

② 計劃期間은 消費地와 消費地의 需要量이 미리 決定되어 있는 單一計劃期間이다.

③ 工場에서 生産되는 製品은 流通設備인 倉庫를 經由해서 消費地에 傳達된다. (工場→倉庫까지의 수송비는 生産工場에서 부담한다.)

④ 각 工場은 資本의 回轉率이 빠르고 必要할 때는 언제든지 殘業을 통해서 製品을 生産하므로 工場의 生産能力은 消費地의 모든 需要를 充足시켜 주는데 充分하다.

⑤ 倉庫의 備蓄能力은 消費地의 모든 需要를 充足시켜 주는데 充分하다.

2-2 用語의 定義

L = 工程의 總 個數

M = 倉庫후보지의 總 個數

N = 消費地의 總 個數

S_{ij} = 工場 i 및 倉庫 j에서 單位製品當 輸送費用(transportation cost)

S_{jk} = 倉庫 j 및 消費地 k에서 單位製品當 輸送費用.

F_j = 倉庫 j의 固定費用(開設, 運營, 維持, 上·下車, 取扱損失費用의 合)

A_i = 計劃期間동안 工場 i가 取扱할 수 있는 製品의 個數의 上限値.

W_j = 計劃期間동안 倉庫 j가 取扱할 수 있는 製品의 個數의 上限値

D_k = 計劃期間동안의 消費地 k의 需要量.

X_{ij} = 工場 i에서 倉庫 j까지 製品의 輸送量.

X_{jk} = 倉庫 j에서 消費地 k까지 製品의 輸送量

$Y_j = 1$ 倉庫 j가 開設(open)될 경우

$Y_j = 0$ 倉庫 j가 閉鎖(close)될 경우

2-3 模型의 定式化

一般的인 2段階 流通시스템에서 倉庫立地選定(Warehouse Location)問題의 定式化는 다음과 같다.

(B) Minimize

$$Z = \sum_{i=1}^{L}\sum_{j=1}^{M} S_{ij}x_{ij} + \sum_{j=1}^{M}\sum_{K=1}^{N} S_{jK}x_{jK} + \sum_{j=1}^{M} F_j Y_j \quad \cdots\cdots (2.1)$$

Subject to

$$\sum_{k=1}^{N} x_{jk} \le W_j Y_j \quad \text{for } (j=1,2,\cdots\cdots M) \cdots (2.2)$$

$$\sum_{i=1}^{L} x_{ij} - \sum_{k=1}^{N} x_{jk} = 0 \quad \text{for } (j=1,2,\cdots\cdots M) \cdots (2.3)$$

$$\sum_{j=1}^{M} x_{jk} = D_k \quad \text{for } (k=1,2,\cdots\cdots,N) \cdots (2.4)$$

$$\sum_{j=1}^{M} x_{ij} \leq A_i \quad \text{for } (i=1,2,\cdots,L) \cdots\cdots (2.5)$$

$$Y_j = 0 \text{ or } 1 \text{ for } (j=1,2,\cdots,M) \cdots\cdots (2.6)$$

$$x_{ij}, x_{jk} \geq 0 \quad \text{for } (i=1,2,\cdots\cdots,L)$$
$$(j=1,2,\cdots\cdots,M)$$
$$(k=1,2,\cdots\cdots,N) \cdots (2.7)$$

그런데 本 硏究에서는 假定 ④에 의거하여 2段階 流通構造에서 倉庫立地選定問題는 적절히 1段階 流通構造內의 倉庫의 立地選定問題로 전환된다.[10]

즉 工場의 生産能力 A_i는 消費地에서 發生하는 모든 需要를 充足시켜 주는데 充分하고 이미 工場은 設立되어 있기 때문에 이와 같은 상황에서는 工場의 生産能力에 關係없이 工場과 倉庫사이에 거리만을 고려하여 줌으로써 倉庫의 最適立地選定을 할 수 있다.

현재 2段階 流通構造問題는 다음과 같이 1段階 流通構造로 전환된다.

(B1) Minimize

$$Z = \sum_{j=1}^{M} \sum_{k=1}^{N} S_{jk}^{*} x_{jk}^{*} + \sum_{j=1}^{M} F_j Y_j$$

Subject to

$$\sum_{k=1}^{N} x_{jk}^{*} \leq W_j Y_j \quad \text{for } (j=1,2,\cdots\cdots,M) \cdots\cdots\cdots (2.8)$$

$$\sum_{j=1}^{M} x_{jk}^{*} = D_k \quad \text{for } (k=1,2,\cdots,N) \cdots\cdots (2.9)$$

$$Y_j = 0 \text{ or } 1 \quad \text{for } (j=1,2,\cdots,M) \cdots (2.10)$$

$$x_{jk}^{*} \geq 0 \quad \text{for } (j=1,2,\cdots\cdots,M), (k=1,2,\cdots, N) \cdots\cdots (2.11)$$

단

$$S_{jk}^{*} = \min S_{ij} + S_{jk}$$

x_{jk}^{*}는 최소비용으로 工場 i에서 倉庫 j를 通해서 消費地에 製品이 配給되는 量.

또 현재 倉庫의 消費能力은 실제로는 製品取扱量에 限界가 있으나 假定 ⑤에 의거하여 관할하는 消費地의 모든 需要를 充足시켜 주는데 充分하므로 위의 定式化는 더욱 더 簡略化된 模型이 될 수 있다.[11]

우선 먼저 다음의 用語를 定義하자.

Z_{jk}는 各 소비지에서 要求하는 需要量 D_k가 倉庫 j에 의해서 供給되는 量의 比率.

즉 $Z_{jk} = x_{jk}^{*} / D_k$

그러므로 假定 ⑤및 위의 定義에 의해서 (B1)모형은 다음의 模型으로 簡略化 된다.

(B2) Minimize

$$Z = \sum_{j=1}^{M} \sum_{k=1}^{N} C_{jk} Z_{jk} + \sum_{j=1}^{M} F_j Y_j \quad \cdots\cdots\cdots\cdots (2.12)$$

Subject to

$$\sum_{j=1}^{M} Z_{jk} = 1 \quad \text{for } (k=1,2,\cdots,N) \cdots\cdots (2.13)$$

$$\sum_{k=1}^{N} Z_{jk} \leq N Y_j \quad \text{for } (j=1,2,\cdots,M) \cdots (2.14)$$

$$Y_j = 0 \text{ or } 1 \quad \text{for } (j=1,2,\cdots,M) \cdots (2.15)$$

$$Z_{jk} \geq 0 \quad \text{for } (j=1,2,\cdots,M), \ (k=1,2,\cdots, N) \cdots\cdots (2.16)$$

단 $C_{jk} = S_{jk}^{*} \times D_k$, $S_{jk}^{*} = \min S_{ij} + S_{jk}$

이 모델은 Z_{jk}값은 連續的인 값을 갖게 되고 Y_j는 0 또는 1의 값을 갖게 되므로 混合二進整数計劃法(0－1 Mixed Integer Programming)의 問題가 된다.

그런데 倉庫의 固定費用을 包含한 混合二進整数計劃法은 計算上 非効率的이며 실제적인 큰 문제에 있어서는 計算의 어려움이 發生한다.

또 消費地의 數나 倉庫의 立地후보지가 無數히 많은 경우는 흔히 볼 수 있으며 이때 倉庫立地選定問題의 計算上의 어려움을 克服하기 위해서는 計算의 効率을 도모할 수 있는 적절한 計算技法이 要求된다.

3. 解를 구하기 위한 알고리즘

3-1 分岐限界法(Branch and Bound Algorithm)의 適用過程

模型(B2)를 해결하기 위해서 本 연구에서 사용하는 技法은 分岐限界法이다.

分岐限界法은 混合整数計劃法을 解決하기 위하여 개발된 方法으로 기본개념은 混合整数計劃法의 整数條件을 제거하고 線形計劃法(Linear Programming)의 解를 차례로 풀어가며 混合整数計劃法의 下限(lower bound)을 증가시켜 가는 방법이다.

3-2 線形計劃法의 簡略化過程

模型에서 Y_j ($j=1, 2, \cdots M$)는 分岐限界法 過程의 각 노드(node)에서 $Y_j=0$ 또는 1의 값을 가진다.

지금 K_0, K_1, K_2를 定義하자.

$K_0=\{j \mid Y_j=0$ 여기서 $j=1, 2, \cdots\cdots, M\}$

$K_1=\{j \mid Y_j=1$ 여기서 $j=1, 2, \cdots\cdots, M\}$

$K_2=\{j \mid$ 0이나 1로 固定되지 않은 Y_j 여기서 $j=1, 2, \cdots\cdots, M\}=(K_0 U K_1)^c$

따라서 임의의 노드에 對應되는 線形計劃法의 定式은 다음과 같다.

Minimize

$$Z=\sum_{j=1}^{M}\sum_{k=1}^{N} C_{jk}Z_{jk}+\sum_{j=1}^{M} F_jY_j \quad \cdots\cdots\cdots\cdots (3.1)$$

Subject to

$$\sum_{j=1}^{M} Z_{jk}=1 \quad \text{for } (k=1,2,\cdots,N) \cdots\cdots \quad (3.2)$$

$$\sum_{k=1}^{N} Z_{jk}\leq 0 \quad \text{for } j\in k_0 \cdots\cdots\cdots\cdots (3.3)$$

$$\sum_{k=1}^{N} Z_{jk}\leq N \quad \text{for } j\in k_1 \cdots\cdots\cdots\cdots (3.4)$$

$$\sum_{k=1}^{N} Z_{jk}\leq NY_j \quad \text{for } j\in k_2 \cdots\cdots\cdots\cdots \quad (3.5)$$

$$Y_j=0 \quad \text{for } j\in k_0 \quad \cdots\cdots\cdots\cdots (3.6)$$

$$Y_j=1 \quad \text{for } j\in k_1 \quad \cdots\cdots\cdots\cdots (3.7)$$

$$0\leq Y_j\leq 1 \quad \text{for } j\in k_2 \quad \cdots\cdots\cdots\cdots (3.8)$$

$$Z_{jk}\geq 0 \quad \text{for } (j=1,2,\cdots,M), (k=1,2,\cdots,N) \cdots\cdots\cdots\cdots (3.9)$$

(3.9)식과 (3.3)식으로부터 모든 K와 집합 K_0에 속하는 모든 j에 대해서 $Z_{jk}=0$이다.

(3.2)식에 의해서 주어진 N개의 制約條件式을 合하면 $\sum_{k=1}^{N}\sum_{j=1}^{M} Z_{jk}=\sum_{k=1}^{N} 1=N$ 이므로 (3.4)식은 重複的(Redundant)이 된다.

또 K_0, K_1, K_2의 定義에 따라서 (3.1)식에서 $\sum_{j=1}^{M} F_jY_j=\sum_{j\in k1} F_j+\sum_{j\in k2} F_jY_j$ 가 된다.

또 $j\in k_0$ 일 때 $Z=_{jk}=0$이 되므로 $\sum_{j=1}^{M}\sum_{k=1}^{N} C_{jk}Z_{jk}=\sum_{j\in k1 V k2}\sum_{k=1}^{N} C_{jk}Z_{jk}$, $\sum_{j=1}^{M} Z_{jk}=\sum_{j\in k1 V k2} Z_{jk}$ 가 된다.

그러므로 (B2)는 $Z=\sum_{j\in k1} F_j+\bar{Z}$ 에서 (B3)를 구하는 문제로 簡略化된다.

(B3) Minimize

$$Z=\sum_{j\in k1 V k2}\sum_{k=1}^{N} C_{jk}Z_{jk}+\sum_{j\in k2} F_jY_j \cdots\cdots\cdots (3.10)$$

Subject to

$$\sum_{j\in k1 V k2} Z_{jk}=1 \quad \text{for } (k=1,2,\cdots,N) \cdots \quad (3.11)$$

$$\sum_{k=1}^{N} Z_{jk}\leq NY_j \quad \text{for } j\in k_2 \quad \cdots\cdots\cdots\cdots (3.12)$$

$$Y_j=0 \quad \text{for } j\in k_0 \cdots\cdots\cdots\cdots (3.13)$$

$$Y_j=1 \quad \text{for } j\in k_1 \cdots\cdots\cdots\cdots (3.14)$$

$$0\leq Y_j\leq 1 \quad \text{for } j\in k_2 \cdots\cdots\cdots\cdots (3.15)$$

$$Z_{jk}\geq 0 \quad \text{for } (j=1,2,\cdots,M), (k=1,2,\cdots,N) \cdots\cdots\cdots\cdots (3.16)$$

위의 定式(B3)에서 Y_j ($j\in k_2$)의 最適解는 $Y_j=\frac{1}{N}\sum_{k=1}^{N} Z_{jk}$가 된다(증명 생략)

이 Y_j를 目的函數값에 대입하면, 目的函數

$$\bar{Z}=\sum_{j\in k1 V k2}\sum_{k=1}^{N} C_{jk}Z_{jk}+\sum_{j\in k2} F_jY_j=\sum_{j\in k1 V k2}\sum_{k=1}^{N} C_{jk}Z_{jk}+\sum_{j\in k2} F_j\cdot\frac{1}{N}\sum_{k=1}^{N} Z_{jk}=\sum_{j\in k1 V k2}\sum_{k=1}^{N} C_{jk}Z_{jk}+\sum_{j\in k2}\sum_{k=1}^{N}$$

$\frac{F_j}{N} Z_{jk}$ 여기서 새로운 記號를 定義하자.

$g_j = 0$ 여기서 $j \in k_1$

$= F_j$ 여기서 $j \in k_2$

그러므로 $\bar{Z} = \sum_{j \in k1 \vee k2} \sum_{k=1}^{N} C_{jk} Z_{jk} + \sum_{j \in k1 \vee k2} \sum_{k=1}^{N} \frac{g_j}{N} Z_{jk}$

$= \sum_{j \in k1 \vee k2} \sum_{k=1}^{N} (C_{jk} + \frac{g_j}{N}) Z_{jk}$

따라서 定式은 다음과 같이 된다.

(B4) Minimize

$$\bar{Z} = \sum_{j \in k1 \vee k2} \sum_{k=1}^{N} (C_{jk} + \frac{g_j}{N}) Z_{jk} \quad \cdots\cdots\cdots\cdots (3.17)$$

Subject to

$$\sum_{j \in k1 \vee k2} Z_{jk} = 1 \quad \text{for } (j = 1, 2, \cdots, M)$$
$$(k = 1, 2, \cdots, N) \quad \cdots\cdots\cdots (3.18)$$

$$Z_{jk} \geq 0 \quad \text{for } j \in k_1 \cup k_2$$
$$(k = 1, 2, \cdots, N) \quad \cdots\cdots\cdots\cdots (3.19)$$

各 k 에 대해서 $\alpha_k = \underset{j \in k1 \vee k2}{\text{Min}} (C_{jk} + \frac{g_j}{N})$ 라고 하면 (B4)의 목적함수값의 下限(lower bound)은 $\bar{Z} = \sum_{k=1}^{N} \alpha_k$ 이므로 $Z = \bar{Z} + \sum_{j \in k1} F_j$ 가 된다.

그러므로 각 노드에서의 線形計劃法의 解는 다음과 같다.

$Z_{jk} = 1$ 여기서 $C_{jk} + \frac{g_j}{N} = \underset{j' \in k1 \vee k2}{\text{Min}} (C_{j'k} + \frac{g_{j'}}{N})$

0 그렇지 않은 경우

$Y_j = 0$ 여기서 $j \in k_0$

$\frac{1}{N} \sum_{k=1}^{N} Z_{jk}$ 여기서 $j \in k_2$

1 여기서 $j \in k_1$

$g_j = F_j$ 여기서 $j \in k_2$

0 여기서 $j \in k_1$

각 노드에서의 線形計劃法의 간략화로 말미암아 計算을 効果的으로 빨리 수행할 수 있다.

3-3 分岐決定規則(Branching Decision Rule)

分岐限界法은 非終結노드(nonterminal node)에서 K_2에 속하는 어떤 j를 선택해서 $Y_j = 1$인 노드와 $Y_j = 0$인 노드로 分岐(branching)하게 되는데 이때 한개 이상의 $j \in k_2$가 존재한다면 어떤 j를 선택하여서 分岐시킬 것인가 하는 문제가 발생한다.

이때 임의로 선택하는 것보다 효율적인 分岐決定規則을 사용함으로써 分岐限界法의 크기와 계산을 상당히 減少시킬 수가 있다.

우선 먼저 다음의 用語를 定義하고 노드의 간략화과정을 설명하면 다음과 같다.

P_j = 倉庫 j에 의해서 供給받을 수 있는 消費地의 集合

N_k = 消費地 k에 供給할 수 있는 倉庫의 集合

$n_j = P_j$의 元素의 個數.

(1) 한 창고를 開設함으로써 얻어지는 最小의 費用節減額(cost saving)을 決定한다.

이때 費用節減額이 그 倉庫를 開設하는데 소요되는 固定費用보다 크다면 그 倉庫는 開設된다.

$$V_{jk} = \min_{\substack{j' \in k1 \vee k2 \\ j' \in Nk \\ j' \neq j}} [\max (C_{j'k} - C_{jk}, 0)] \quad j \in k_2, k \in p_j$$

$$\triangle_j = \sum_{k \in P_j} V_{jk} - F_j$$

이때 $\triangle_j > 0$라면 $Y_j = 1$이 된다.

(2) n_j를 감소시키는 방법

$j \in k_2$, $k \in P_j$ 에 대해서

$$\min_{\substack{j' \in k1 \\ j' \in Nk}} (C_{j'k} - C_{jk}) < 0$$

이라면 n_j는 1만큼 감소된다.

(3) 이미 開設된 倉庫 $j (j \in k_1)$에 의해서 供給하지 않고 그 이외의 倉庫를 開設하여 얻을 수 있는 最大費用節減額(maximum cost saving)을 決定한다.

이때 費用節減額이 그 倉庫를 開設하는데 所要되는 固定費用보다 작다면 倉庫는 閉鎖된다.

$$W_{jk} = \min_{\substack{j' \in k1 \\ j' \in Nk}} [\max (\bar{C}_{j'k} - C_{jk}, 0)] \quad j \in k_2, k \in P_j$$

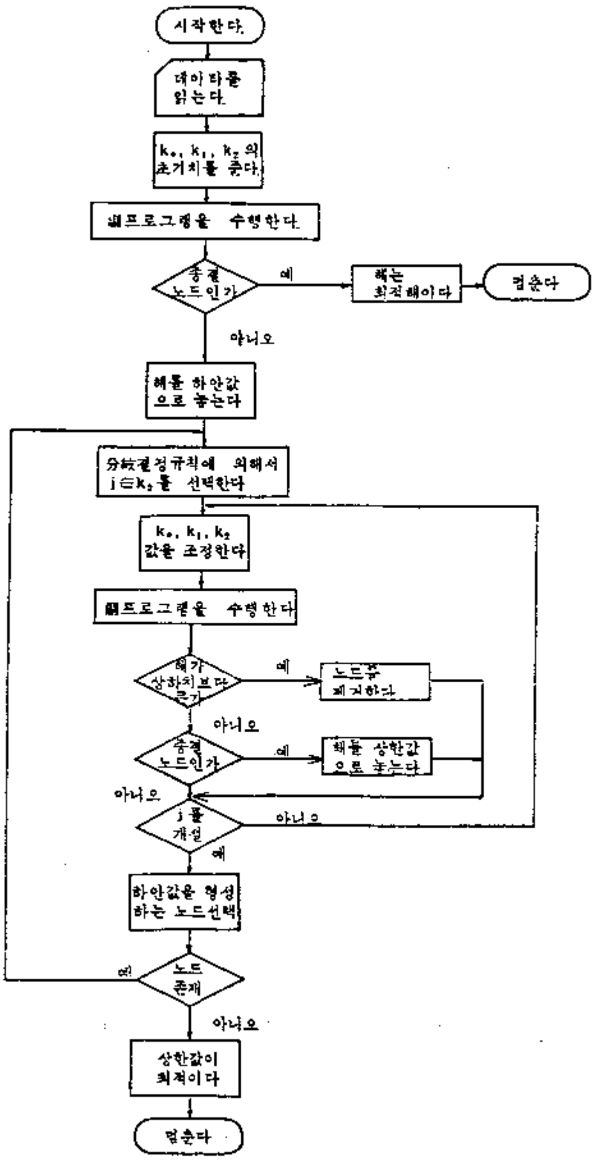
$$\Omega_j = \sum_{k \in P_j} W_j - F_j, \quad j \in k_2$$

이때 $\Omega_j < 0$라면 $Y_j = 0$이 된다.

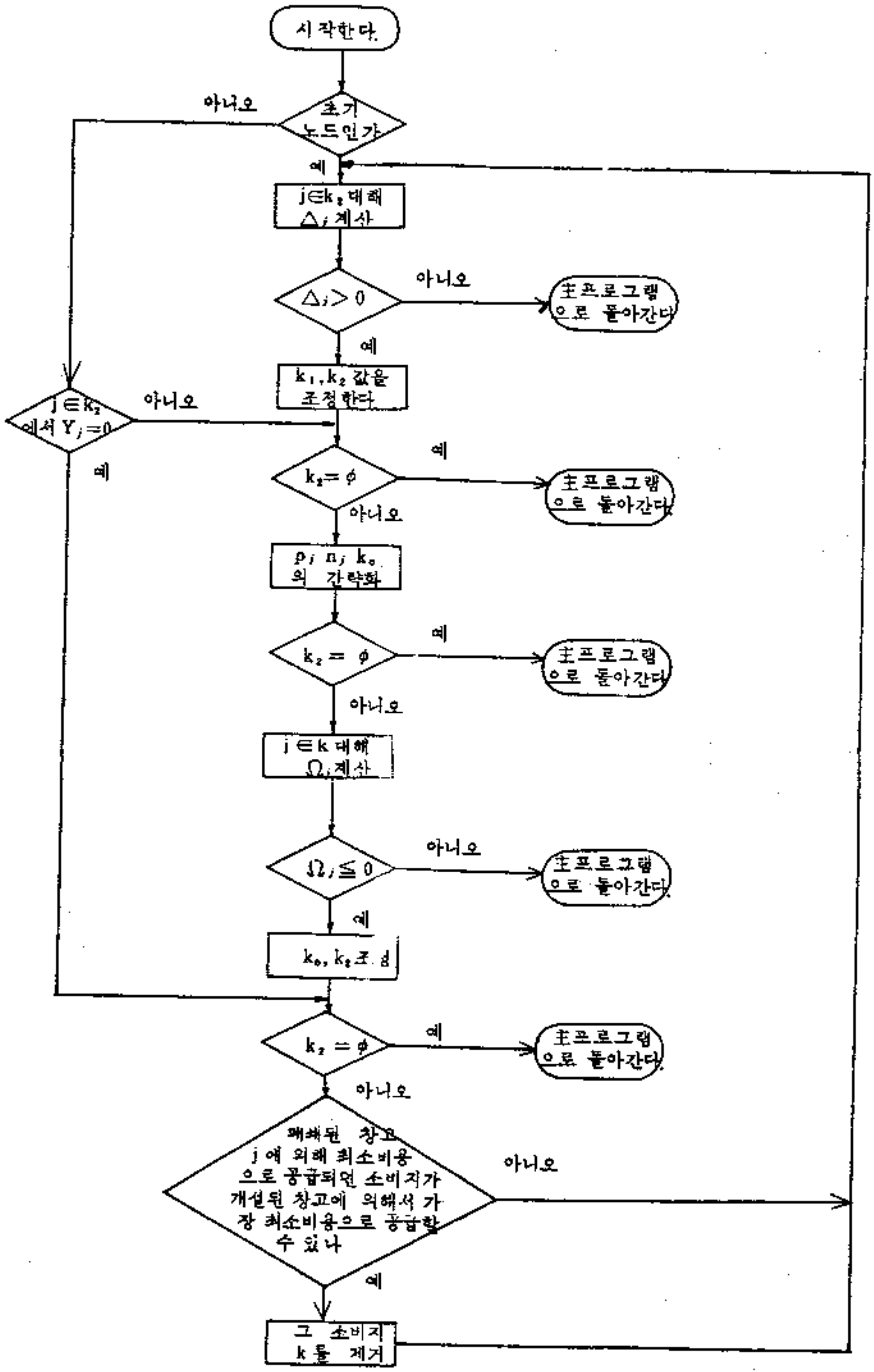
本 硏究에서는 위의 노드 간략화 과정을 수행하는 동안 計算된 0보다 큰 Ω_j값 중에서 가장 큰 j값을 선택하는 것을 分岐決定規則으로 선택하고 있으며 그 j에 $Y_j = 0$과 $Y_j = 1$ 값을 주어 分岐(branching)시키고 있다.

3—4 알고리즘(Algorithm)의 設計

倉庫의 立地選定問題를 해결하기 위한 간략화된 分岐限界法 節次의 흐름圖(Flow Chart)는 그림1 및 그림2와 같다.



(그림1) 主프로그램(Main program)의 흐름 図 (Flow Chart)



(그림 2) 副프로그램 (Sub program) 의 흐름図 (Flow Chart)

4. 結　　論

企業의 장기적인 意思決定問題로서 重要한 比重을 차지하는 流通시스템 (Distribution System)을 決定할 때 流通設備가 되는 工場이나 倉庫의 立地選定은 대단히 重要하며 流通시스템의 總流通費를 最小化시키는 合理的인 方法이 요구된다.

本 硏究에서는 現實적으로 흔히 볼 수 있는 2段階 流通시스템의 네트워크上의 立地選定(Location on a Network)문제일 경우 流通設備인 倉庫의 立地選定(Warehouse Location) 모형을 混合二進整数計劃法(0−1 Mixed Integer Programming)으로 정식화하고, 이를 1段階 流通構造로 전환시켰다.

이 模型을 効率的으로 계산하기 위해서 分岐限界法(Branch and Bound Algorithm)을 이용하고 그 과정에서 線形計劃法(Linear Programming)의 간략화와 가장 큰 Ω값을 선택하는 分岐決定規則(Branch Decision Rule)을 통해서 解를 경제적으로 구하였다.

그리고 最適解를 구하기 위한 알고리즘을 設計하고 이것을 FORTRAN 프로그램으로 코딩(coding)하였으나 컴퓨타 프로그램의 提示는 생략하였다.

Reference

1. Kuehn, A. A. and M. J. Hamburger, "A Heuristic Program for Locating Warehouses," *Management Science*, Vol. 9, No. 4 July 1963, pp. 643-666.
2. Revelle, C., D. Marks, and J. Liebman, "An Analysis of Private and Public Sector Location Models," *Management Science*, Vol. 16, No. 11, July 1970, pp. 692-708.
3. Geoffrion, A. M., "A Guide to Computer-Assisted Methods for Distribution System Planning," *Sloan Management Review*, Vol. 16, No. 2, Winter 1975, pp. 17-41.
4. Geoffrion, A. M. and G. W. Graves, "Multicommodity Distribution System Design by Benders Decomposition," *Management Science*, Vol. 20, No. 5, January 1974, pp. 822-844.
5. Efroymson, M. A. and T. L. Ray, "Branch-Bound Algorithm for Plant Location," *Operation Research*, Vol. 14, No. 3, May-June 1966, pp. 361-368.
6. Atkins, R. L. and R. H. Shriver, "New Approach to Facility Location," *Harvard Business Review*, May-June 1968, pp. 70-79.
7. Khumawala, B. W., "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem," *Management Science*, Vol. 22, No. 1, September 1975, pp. 51-56.
8. Davis, P. S. and T. L. Ray, "A Branch-Bound Algorithm for the Capacitated Facilities Location Problem," *Naval Research Logistics Quarterly*, Vol. 16, No. 3, November 1969, pp. 331-343.
9. L. Kauman, and M. V. Eede, "A Plant and Warehouses Location Problem," *Operation Research Quarterly*, Vol. 28, No. 3, 1977, pp. 547-554.
10. Ellwein, L. B. and P. Gray, "Solving Fixed Charge Location-Allocation Problems with Capacity and Side Constraints," *AIIE Transaction*, Vol. 3, No. 4, 1971, pp. 290-298.
11. Francis, R. L., *Facility Layout and Location*, Prentice Hall, Inc., 1974.
12. Balinski, M. L., and H. Mills, "A Warehouse Problem," Prepared for the Veterans Administration, *Mathematica*, Princeton, N. J., April 1960.
13. Wagner, H. M., *Principles of Operation Research*, Prentice Hall, 1969.