〈研究論文〉

Journal of the Korean Institute of Industrial Engineers
Vol. 11, No. 1, June 1985

2次計劃法을 利用한 備蓄面積의 制限을 받는 多品種注文量決定시스템에 關한 硏究.

(A Study of the Multi－Item Order－Level System with Storage Limitation by the Quadratic Programming)

姜 東 鎭*
李 相 鎔**

Abstract

This paper analyzes the multi-item order-level system with shortages allowance and storage limitation. Up to now, we have used the classical optimization theory to analyze this system. But the theory is generally not suitable for computational purposes. Therefore, this paper designes a new method to be able to apply the quadratic programming to the multi-item order-level system with storage limitation. A numerical example is also presented.

1. 序 論

一般的인 多品種 在庫管理 模型은 偏微分法等으로 쉽게 分析할 수 있으나 制限式이 생기는境遇의 이 模型은 分析이 容易하지 않으며, 在來的인 最適化理論(Classical Optimization Theory)을 利用하는 方法[1][2] 밖에는 그 解法이 없다. 그런데 이 方法은 品種이나 制限式이 多数인 境遇에는 解法이 대단히 複雜하여 實用性이 없다.

本 硏究는 備蓄面積에 대한 制限式이 있는 境遇의 品切損失이 許容되는 多品種注文量 決定시스템(Multi-Item Order-Level System)에 關한 解法의 設計이다. 品切損失이 許容되는 多品種注文量決定시스템은 그 費用函数가 注文量에 대

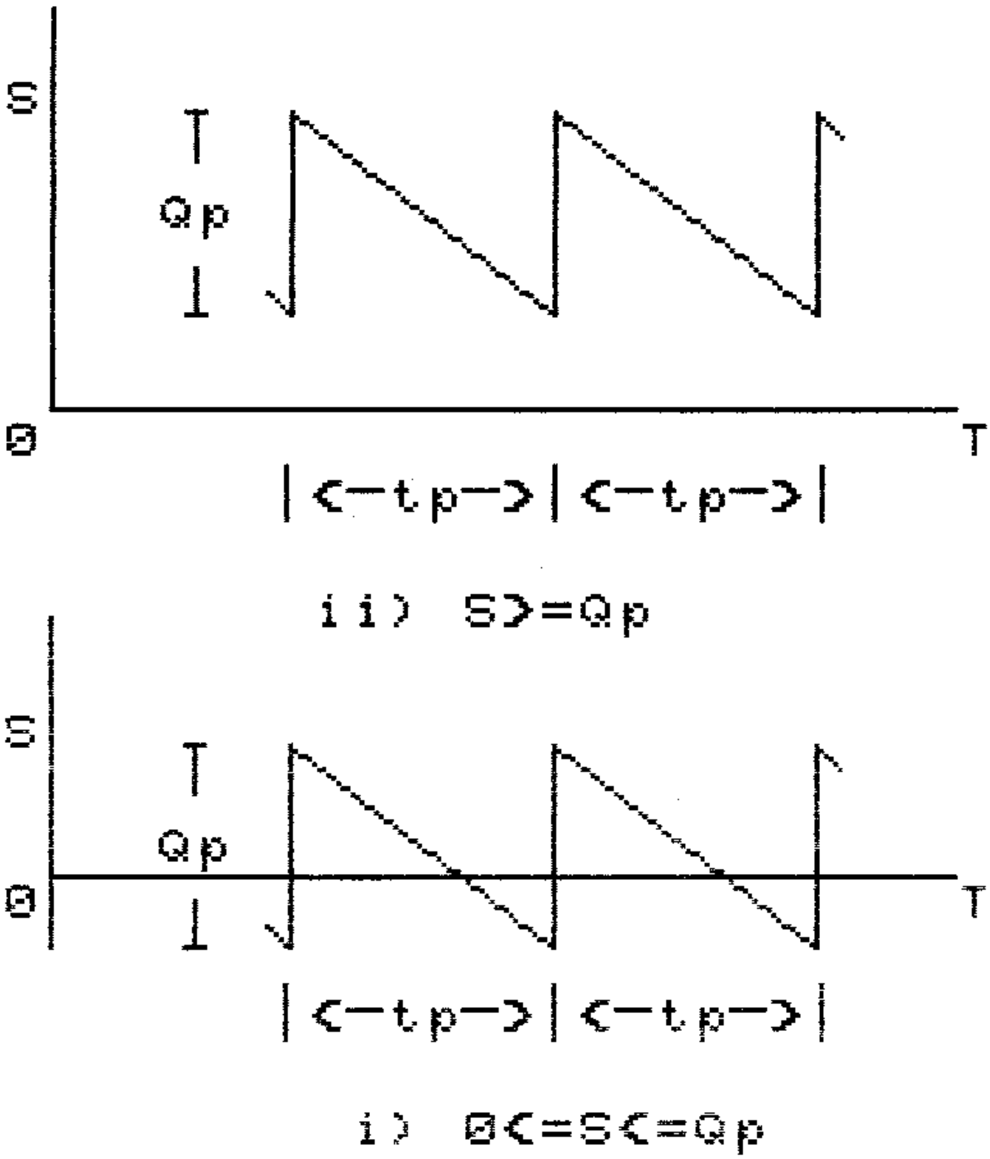
*建國大學校 産業工學科 大學院
**建國大學校 産業工學科 教授

한 2次式이고 備蓄面積에 대한 制限式이 線型이므로 2次計劃法(Quadratic Programming)의 解法을 利用할 수 있다.

2次計劃法의 解法으로는 여러 가지가 있으나[3][4][5][6][7] 本 硏究에서는 쿤-타카 條件式(Kuhn-Tucker Conditions)을 利用하여 目的函數를 線型으로 變換시키고 새로운 制限式을 附加하여 심플렉스 方法(Simplex Method)化 한 울프의 解法(Wolf Algorithm)을 利用하였다.[3][8][9]

그리고, 實際로 可用備蓄面積이나 運轉資金 等의 制限을 받는 購買者가 小型컴퓨터(Mini/Micr Computer)로도 쉽게 利用할 수 있도록 베이직(BASIC)으로 코-딩(Coding)함으로써 實用性을 增大시켰다.

2. 2次計劃法(Quadratic Programming)을 利用한 備蓄面積의 制限이 있는 多品種 注文量決定시스템(Multi-Item Order-Level System) 分析.



〈그림 1〉 Order-Level System

(1) 注文量決定시스템(Order-Level System) [2]

〈그림 1〉은 注文量決定시스템의 一般的인 形態를 나타낸 것인데, 이를 分析하기 위하여 記号들을 다음과 같이 定義한다.

TC＝(增分)総費用(원/期間)

q_P ＝로트의 크기(1回分 生産量)

C_1 ＝在庫維持保管費用(원/單位/期間)

C_2 ＝品切損失費用(원/單位/期間)

S ＝注文量

t_P ＝計劃期間(Scheduling Period)

I_1 ＝平均在庫量(單位/期間)

I_2 ＝平均在庫量 품절양

〈그림 1〉을 分析해 보면, 平均在庫量은 式(1)(2)와 같고, 計劃期間동안 豫想되는 平均品切量은 式(3),(4)와 같다.

$$I_1 = \begin{cases} S^2/2q_P, & 0 \leqq S \leqq q_P \quad \cdots\cdots (1) \\ S - q_P/2, & q_P \leqq S \quad \cdots\cdots (2) \end{cases}$$

$$I_2 = \begin{cases} (q_P - S)^2/2q_P, & 0 \leqq S \leqq q_P \quad \cdots\cdots (3) \\ 0, & q_P \leqq S \quad \cdots\cdots (4) \end{cases}$$

따라서 総費用은 式(5),(6)이 된다.

$$TC = \begin{cases} C_1 S^2/2q_P + C_2 (q_P - S)^2/2q_P, & 0 \leqq S \leqq q_P \quad \cdots\cdots (5) \\ C_1 (S - q_P/2), & q_P \leqq S \quad \cdots\cdots (6) \end{cases}$$

그런데 本 硏究에서는 〈그림 1〉의 첫째 模型만을 取扱하기로 하겠다.

(2) 備蓄面積의 制限을 받는 多品種模型(Multi-Item Model with Storage Limitation)

〈그림 1〉의 첫째 模型에서 多品種일 境遇, i 번째 品種의 単位当 로트의 크기를 q_{Pi}, 単位当 在庫維持保管費를 C_{1i}, 単位当 品切損失費를 C_{2i}, 単位当 注文量을 S_i라 했을 때, 式(5)는 다음의 式(7)로 擴張된다.

$$TC(S_1, S_2, \cdots\cdots S_N) = \sum_{i=1}^{N} \frac{C_{1i} S_i^2 + C_{2i}(q_{Pi} - S_i)^2}{2q_{Pi}} \quad \cdots\cdots (7)$$

그리고 N品種에 대한 最大可用備蓄面積을 B라고 하고 i번째 品種의 単位当 所要備蓄面積을 b_i라 한다면 備蓄面積에 대한 制限式은 式(8)이 된다.

$$\sum_{i=1}^{N} b_i s_i \leqq B \quad \cdots\cdots (8)$$

따라서, 備蓄面積의 制限이 있고 品切損失이 許容되는 多品種注文量決定시스템은 다음의 式(9),(10),(11),(12)로 数式化(Formulation)된다.

目的函数 : Minimize

$$TC(S_1, S_2, \cdots\cdots S_N)$$

$$= \sum_{i=1}^{N} \frac{C_{1i} S_i^2 + C_{2i}(q_{Pi} - s_i)^2}{2q_{Pi}} \quad \cdots\cdots (9)$$

制限式 : Sbject to

$$\sum_{i=1}^{N} b_i s_i \leqq B \quad \cdots\cdots (10)$$

$$S_i \leqq q_{Pi} \quad \cdots\cdots (11)$$

$$S_i \geqq 0 \quad \cdots\cdots (12)$$

(3) 2次計劃法(Quadratic Programming)에 의한 解法.

가. 2次計劃法의 基本型을 다음의 式 (13),(14),(15)로 定義한다.

目的函数 : Minimize $\quad Z = CX + X^T QX \quad \cdots (13)$

制限式 : Sbject to $\quad AK \leqq B \quad \cdots\cdots (14)$

$\quad X \leqq O \quad \cdots\cdots (15)$

여기서

$$X = [X_1, X_2, \cdots\cdots X_N]^T$$

$$C = [C_1, C_2, \cdots\cdots C_N]$$

$$B = [b_1, b_2, \cdots\cdots b_M]^T$$

$$A = \begin{bmatrix} a_{11}, a_{12}, \cdots\cdots a_{1N} \\ a_{12}, a_{22}, \cdots\cdots a_{2N} \\ \vdots \quad \vdots \quad \vdots \\ a_{M1}, a_{M2}, \cdots\cdots a_{MN} \end{bmatrix}$$

$$Q=\begin{bmatrix} q_{11}, q_{12}, \cdots\cdots\cdots q_{1N} \\ q_{21}, q_{22}, \cdots\cdots\cdots q_{2N} \\ \vdots \quad \vdots \quad\quad \vdots \\ q_{N1}, Q_{N2}, \cdots\cdots\cdots q_{NN} \end{bmatrix}$$

N =变数의 数

M =制限式의 数

Q =对称行例(Symmetric Matrix)

그리고 式(9), (10), (11), (12)를 式(13), (14), (15) 에서 定義한 2次計劃法(Quadratic Programming)의 基本型으로 나타내면 다음과 같다.

式(9)에서

Minimize

$$TC\ (S_1, S_2, \cdots\cdots S_N)=\sum_{i=1}^{N} \frac{C_{1i}S_i^2+C_{2i}(q_{Pi}-S_i)^2}{2q_{Pi}}$$

$$=\sum_{i=1}^{N} \frac{C_{1i}S_i^2+C_{2i}S_i^2-2C_{2i}q_{Pi}S_i+C_{2i}q_{Pi}^2}{2q_{Pi}}$$

$$=-\sum_{i=1}^{N} C_{2i}S_i+\sum_{i=1}^{N}\frac{(C_{1i}+C_{2i})}{2q_{Pi}}S_i^2+\sum_{i=1}^{N}\frac{C_{2i}}{2}q_{Pi}$$

$$=-C_{21}S_1-C_{22}S_2\cdots\cdots\cdots\cdots C_{2N}S_N$$

$$+\frac{C_{11}+C_{21}}{2q_{P1}}S_1^2+\frac{C_{12}+C_{22}}{2q_{P2}}S_2^2+\cdots\cdots\cdots$$

$$\frac{C_{1N}+C_{2N}}{2q_{PN}}S_N^2+\delta$$

$$=[-C_{21}, -C_{22}, \cdots\cdots, -C_{2N}]\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{bmatrix}$$

$$+[S_1, S_2\cdots S_N]\begin{pmatrix} \frac{(C_{11}+C_{21})}{2q_{P1}}, & 0, & \cdots 0 \\ 0 & \frac{(C_{12}+C_{22})}{2q_{P2}} & \cdots 0 \\ \vdots & \vdots & \vdots \\ 0, & 0, & \frac{(C_{1N}+C_{2N})}{2q_{PN}} \end{pmatrix}\begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{pmatrix}+\delta$$

그런데, Minimize $TC(S_1, S_2, \cdots\cdots S_N)$ =Minimize $(TC\ (S_1, S_2, \cdots\cdots S_N)+\delta)$ 이므로 Minimize $(TC\ (S_1, S_2, \cdots\cdots S_N)-\delta)$ =Minimize Z 라고 假定한다.

결국

$$\text{Minimize } Z=[-C_{21}, -C_{22}, \cdots\cdots -C_{2N}]\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{bmatrix}$$

$$+[S_1, S_2, \cdots\cdots S_N]\begin{bmatrix} \frac{(C_{11}+C_{21})}{2q_{P1}}, & 0 & , \cdots\cdots 0 \\ 0 & , \frac{(C_{12}+C_{22})}{2q_{P2}} & , \cdots\cdots 0 \\ 0 & , \quad 0 & , \frac{(C_{1N}+C_{2N})}{2q_{PN}} \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{bmatrix}$$

$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (16)$$

이 된다.

式(10),(11),(12)에서

$$\begin{bmatrix} b_1, b_2, \cdots\cdots\cdots b_N \\ 1\ ,0\ , \cdots\cdots\cdots 0 \\ 0\ ,1\ , \cdots\cdots\cdots 0 \\ \vdots \quad \vdots \quad\quad \vdots \\ 0\ ,0\ , \cdots\cdots\cdots 1 \end{bmatrix}\begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_N \end{bmatrix} \leqq \begin{bmatrix} B \\ q_{P1} \\ q_{P2} \\ \vdots \\ q_{PN} \end{bmatrix} \cdots (17)$$

$$S_i \leqq 0,\ \ i=1, 2, \cdots\cdots\cdots N \cdots\cdots\cdots\cdots\cdots \quad (18)$$

이 된다.

따라서 式 (9), (10), (11), (12)로 表示된 可用備蓄面積의 制限이 있는 多品種注文量決定시스템(Multi-Item Order-Level System)은 式 (16), (17), (18)과 같은 一般的인 2次計劃法(Quadratic Programming)問題로 된다.

나. 2次計劃法의 解法

式 (13), (14), (15)를 다음의 式(19), (20), (21)로 變型시키면

$$\text{Minimize}\quad Z=\sum_{j=1}^{N} C_j x_j+\sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk}x_j x_k$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (19)$$

$$\text{Subject to}\ \ \sum_{j=1}^{N} a_{ij}x_j-b_i \leqq 0,\ \ i=1, 2,$$
$$\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (20)$$

$$-x_j \leqq 0,\ \ j=1, 2, \cdots\cdots N \cdots\cdots\cdots\cdots\cdots\cdots (21)$$

$$\sum_{j=1}^{N} a_{ij}x_j+S_i \leqq b_i,\ \ i=1, 2, \cdots\cdots M \cdots\cdots (20')$$

이 된다.

그리고 쿤-타카 條件式(Kuhn-Tucker Conditions)

$$\lambda \leqq 0$$
$$\nabla f(X) + \lambda \nabla g(X) = 0$$
$$\lambda_i g_i(x) = 0, \quad i = 1, 2, \cdots\cdots M$$
$$g(X) \leqq 0$$

으로부터

$$C_j + 2\sum_{k=1}^{N} q_{jk}x_k + \sum_{i=1}^{M} a_{ij}\lambda_i - \lambda_{M+j} = 0 \quad j=1,2,\cdots N \cdots (22)$$

$$\lambda_i \left(\sum_{j=1}^{N} a_{ij}x_j - b_i\right) = 0, \quad i=1,2,\cdots M \cdots (23)$$

$$\lambda_{M+j}(-x_j) = 0, \quad j=1,2,\cdots N \cdots (24)$$

이 된다.

그런데 $\lambda_{M+j} = W_j$라 한다면 式(22), (23), (24)는

$$2\sum_{k=1}^{N} q_{jk}x_k + \sum_{i=1}^{M} a_{ij}\lambda_i - W_j = -C_j, \quad j=1,2,\cdots\cdots N \cdots\cdots (25)$$

$$\lambda_i S_i = 0, \quad i=1,2,\cdots\cdots M \cdots\cdots (26)$$

$$W_j x_j = 0, \quad j=1,2,\cdots\cdots N \cdots\cdots (27)$$

이 되고, S_i, λ_i, w_j, x_j는 모두 非陰数(Non-negative)이기 때문에 式(26), (27)을 함께 나타내면 式(28)이 된다.

$$\sum_{i=1}^{M} \lambda_i x_i + \sum_{j=1}^{N} w_j x_j = 0 \cdots\cdots (28)$$

그리고 式(20′), (25), (28)로부터 式(29)가 誘導되고

$$2\sum_{j=1}^{N}\sum_{k=1}^{N} q_{jk}x_k x_j = -\sum_{i=1}^{N} b_i\lambda_i - \sum_{j=1}^{N} C_j x_j \cdots\cdots (29)$$

式(19)는 式(29)에 의하여 式(30)으로 된다.

$$Z = \frac{1}{2}\left(\sum_{j=1}^{N} C_j x_j - \sum_{i=1}^{M} b_i\lambda_i\right) \cdots\cdots (30)$$

따라서 式(13), (14), (15)의 2次計劃法 模型은 目的函数가 式(30)이고 制限式이 式(20′) (25)인 線型計劃法(Linear Programming) 模型이 된다.

이제 初期 심플렉스表(Initial Simplex Tableu)를 〈表 1〉과 같이 設計하고, 이를 심플렉스 方法(Simplex Method)으로 反復(Iteration) 하면 最適解를 구할 수 있다.

3. 数値例

品切損失이 許容되는 注文量決定시스템(Order-Level System)에서 品種이 세가지, 可用 備蓄面積의 2000(ft³)이라 한다. 그리고〈表 2〉와 같이 必要한 資料가 주어질 때 各品種의 最適注文量을 決定하라.

〈表 1〉 Initial Simplex Tableau.

	X (J)	1,2,... N,	N+1,... N+K,	N+K+1,... 2N+K,	2N+K+1,... 2N+2K,	2N+2K+1,... 3N+2K
C (I)	C (J)	O	B	O	O	M
O	B	A	O	O	I	O
M	–C	2Q	A	–I	O	I
Z (J)	–MC	2MQ	MA	–M	O	M
C(J) – Z(J)		–2MQ	B–MA	M	O	O

*M = Big M

〈表 2〉 **Data**

品 種	i	1	2	3
로트의 크기	q_{Pi}	200	100	500
在庫管理維持費	C_{1i}	2.0	1.0	4.0
品切損失費	C_{2i}	50.0	40.0	20.0
可用面積 (ft^3)	b_i	5.0	3.0	9.0

위 事例의 数学的 模型은 式(16), (17), (18)에 의하여

$$\text{Minimize} \quad Z = [-50, -40, -20] \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix} + [S_1, S_2, S_3] \begin{bmatrix} 0.13, & 0, & 0 \\ 0, & 0.205, & 0 \\ 0, & 0, & 0.024 \end{bmatrix} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \end{bmatrix}$$

〈表 3〉 Initial Simplex Tableau

(Iteration 0)

	X (J)	1	2	3	4	5	6	7	8
		9	10	11	12	13	14	15	16
		17							
C(I)	C(J)	0.00	0.00	0.00	2000.00	200.00	100.00	500.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	9999.00	9999.00
		9999.00							
0.00	2000.00	5.00	3.00	9.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
		0.00							
0.00	200.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
		0.00							
0.00	100.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
		0.00							
0.00	500.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
		0.00							
9999.00	50.00	0.26	0.00	0.00	5.00	1.00	0.00	0.00	-1.00
		0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00
		0.00							
9999.00	40.00	0.00	0.41	0.00	3.00	0.00	1.00	0.00	0.00
		-1.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00
		0.00							
9999.00	20.00	0.00	0.00	0.05	9.00	0.00	0.00	1.00	0.00
		0.00	-1.00	0.00	0.00	0.00	0.00	0.00	0.00
		1.00							
Z (J)	*******	2599.74	4099.59	479.95	*******	9999.00	9999.00	9999.00	-9999.00
		-9999.00	-9999.00	0.00	0.00	0.00	0.00	9999.00	9999.00
		9999.00							
C(J) - Z(J)		-2599.74	-4099.59	-475.95	*******	-9799.00	-9899.00	-9499.00	9999.00
		9999.00	9999.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00							

$$\text{Subject to} \begin{pmatrix} 5, & 3, & 9 \\ 1, & 0, & 0 \\ 0, & 1, & 0 \\ 0, & 0, & 1 \end{pmatrix} \begin{pmatrix} S_1 \\ S_2 \\ S_3 \end{pmatrix} \leq \begin{pmatrix} 2000 \\ 200 \\ 100 \\ 500 \end{pmatrix}$$

$$S_i \geq 0, \quad i = 1, 2, 3$$

이 된다.

그리고 위의 事例를 베이직 프로그램(BASIC Program)에 依據하여 마이크로컴퓨터(Micro Computer)로 計算한 結果는 〈表3〉, 〈表4〉, 〈表5〉, 〈表6〉과 같다.

〈表 4〉

(Iteration 0)

	X (J)	1	2	3	4	5	6	7	8
		9	10	11	12	13	14	15	16
		17							
C(I)	C(J)	0.00	0.00	0.00	2000.00	200.00	100.00	500.00	0.00
		0.00	0.00	2.22	0.00	0.00	0.00	9999.00	9999.00
		9999.00							
2.22	2000.00	5.00	3.00	9.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
		0.00							
0.00	200.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00
		0.00							
0.00	100.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	1.00	0.00	0.00	0.00
		0.00							
0.00	500.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	1.00	0.00	0.00
		0.00							
9999.00	38.89	0.26	0.00	−0.03	0.00	1.00	0.00	−0.56	−1.00
		0.00	0.56	0.00	0.00	0.00	0.00	1.00	0.00
		−0.56							
9999.00	33.33	0.00	0.41	−0.02	0.00	0.00	1.00	−0.33	0.00
		−1.00	0.33	0.00	0.00	0.00	0.00	0.00	1.00
		−0.33							
2000.00	2.22	0.00	0.00	0.01	1.00	0.00	0.00	0.11	0.00
		0.00	−0.11	0.00	0.00	0.00	0.00	0.00	0.00
		0.11							
Z(J)	*******	2610.85	4106.26	395.96	2000.00	9999.00	9999.00	−8665.78	−9999.00
		−9999.00	8665.78	2.22	0.00	0.00	0.00	9999.00	9999.00
		−8665.78							
C(J) − Z(J)		−2610.85	−4106.26	395.96	0.00	−9799.00	−9899.00	9165.78	9999.00
		9999.00	−8665.78	0.00	0.00	0.00	0.00	0.00	0.00
		*******							

(Iterations 2 through 5 are omitted.)

〈表 5〉 (Iteration 6)

	X (J)	1	2	3	4	5	6	7	8
		9	10	11	12	13	14	15	16
		17							
C(I)	C(J)	0.00	0.00	0.00	0.00	39.69	14.61	395.30	160.31
		85.39	104.70	1.66	0.00	0.00	0.00	9999.00	9999.00
		9999.00							
0.00	104.70	0.00	0.00	1.00	0.00	–2.00	–0.76	1.36	2.00
		0.76	–1.36	0.10	0.00	0.00	0.00	–2.00	–0.78
		1.36							
0.00	39.69	0.00	0.00	0.00	0.00	–3.64	0.08	2.00	3.64
		–0.08	–2.00	–0.01	1.00	0.00	0.00	–3.64	0.08
		2.00							
0.00	14.61	0.00	0.00	0.00	0.00	0.08	–2.41	0.76	–0.08
		2.41	–0.76	–0.00	0.00	1.00	0.00	0.08	–2.41
		0.76							
0.00	395.30	0.00	0.00	0.00	0.00	2.00	0.76	–1.36	–2.00
		–0.76	1.36	–0.10	0.00	0.00	1.00	2.00	0.76
		–1.36							
0.00	160.31	1.00	0.00	0.00	0.00	3.64	–0.08	–2.00	–3.64
		0.08	2.00	0.01	0.00	0.00	0.00	3.64	–0.08
		–2.00							
0.00	85.39	0.00	1.00	0.00	0.00	–0.08	2.41	–0.76	0.08
		–2.41	0.76	0.00	0.00	0.00	0.00	–0.08	2.41
		–0.76							
0.00	1.66	0.00	0.00	0.00	1.00	0.01	0.00	0.10	–0.01
		–0.00	–0.10	–0.00	0.00	0.00	0.00	0.01	0.00
		0.10							
Z(J)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
		0.00							
C(J) – Z(J)		0.00	0.00	0.00	0.00	39.69	14.61	395.30	160.31
		85.39	104.70	1.66	0.00	0.00	0.00	9999.00	9999.00
		9999.00							

〈表 6〉 Program Output

The # of iteration : 6

THE MINIMUM VALUE Z : –8426.32
Constant : 12000
Total cost : 3573.68

Numbers 1 through 3 are ordinary variables.
Numbers 4 through 7 are Lagrangians 1 (Lambda j).
Numbers 8 through 10 are Lagrangians 2 (Wj).
Numbers 11 through 14 are slacks.
Numbers 15 through 17 are artificials.

The optimum points are printed below.

X (3) = 105
X (12) = 40
X (13) = 15
X (14) = 395
X (1) = 160
X (2) = 85
X (4) = 2

The rest of the variables are equal to zero.

4. 結　論

多品種 決定的 在庫管理 模型(Multi-Item Deterministic Model)을 分析하는데 있어서 지금까지는 一般的으로 在來的인 最適化理論(Classical Optinuization Theory)을 利用하였다. 그러나 여러 가지 制約條件이 있는 境遇에는 이 理論의 適用이 상당히 複雜할 뿐만 아니라 實用性이 적었다.

그런데, 品切損失이 許容되는 多品種 模型에서 備蓄面積 等의 制限을 받는 境遇에 2次計劃法의 解法이 適用될 수 있으며 이는 電算化에도 便利하여 그 實用性이 크게 向上되리라는 생각이다.

本 硏究에서는 備蓄面積의 制限을 받는 境遇의 品切損失이 許容되는 多品種 注文量決定시스템을 分析하고 있으나, 運轉資金의 制限을 받는 境遇, 혹은 이들의 制限을 동시에 받는 境遇는 물론 유사한 制約條件下의 多品種生産量決定시스템(Multi-Item Production-Level System)에도 이 方法이 適用될 수 있다.

그러나, 費用函數가 2次式이 아닌 一般的인 多品種模型에서 여러 形態의 制限式이 있는 境遇에는 다른 高等非線型計劃法(Advanced Techniques of Nonlinear Programming) 等의 解法으로 分析이 가능하리라 생각되며 이 分野의 硏究 또한 本 硏究의 연장으로 계속 수행되어야 하겠다.

References

1. Taha, H., *Operations Research*, 3nd 3d., pp. 504-507, Macmillan Publishing Co., Inc., New York, 1982.
2. Naddor, Eliezer, *Inventory Systems*, Ch. 4, John Wiley & Sons, Inc., New York, 1966.
3. Wolf, P.,"The Simplex Method for Quadratic Programming" *Econometrica*, Vol. 27, pp. 382-389, 1959.
4. Beale, E.M.L.,"On Quadratic Programming," *Nevel Research Logistics Quarterly*, Vol. 6, pp. 227- 243, 1959.
5. Theil, H., and C. Van der Panne,"Quadratic Programming as an Extension of Classical Maximization," *Management Science*, Vol. 7, pp. 1-20, 1960.
6. Lemke, C.E., "A Method of Solution for Quadratic Programmings," *Management Science*, Vol. 8, pp442-453, 1962.
7. Braitsch, R.J. Jr., "A Computer Comparison of Four Quadratic Programming Algorithms," *Management Science*, Vol. 18, pp. 632-643, 1972.
8. Wagner, H.M., *Principles Operations Research*, 2nd ed., pp. 612-621, Prentice-Hall, Inc., Englewood Cliffs, 1975.
9. Simmons, D.M., *Nonlinear Programming for Operations Research*, Ch. 7, Prentice-Hall, New York, 1975.