□ 技 術 解 說 □

Linear Programming技法에 있어서 Basis Matrix의 Bartels-Golub Decomposition 및 그 응용

文 永 鉉*

■ 차 례 ■

1. LP에 關聯된 Dimensionality 問題
2. B-G Decomposition에 의한 Basis Matrix의 Factor Table 修正Algorithm
3. 修正된 Factor Table에 의한 dual cost vector 計算 및 reduced cost vector의 updating
4. B-G Algorithm의 変種(variants)
5. B-G Algrithm의 應用事例 및 檢討
參 考 文獻

大規模 system 을 取扱하는 工學分野에 있어서 Matrix의 demension이 커짐에 따라 電算負擔을 줄이기 위한 所要記憶容量의 節減 및 計算時間의 短縮이 주된 관심사 중의 하나가 되어 왔다. LP(Linear Programming) 技法 역시 대규모 線形 system에 適用될 경우 Basis Matrix에 대한 dimensionality 問題를 포함하고 있으며 이에 대한 効果的인 解決方法으로 Bartels - Golub Decomposition 方法이 研究되어 있다.

이 力法은 Chan 과 Yip 等이 LSGA (Load Shedding and Generator Rescheduling) 問題에의 適用을 試圖한 바 있으며, 一般的인 線形最適化問題뿐만 아니라, dimensionality 가 큰 Matrix 의 Inverse 計算을 要하는 問題에 널리 適用될 수 있으므로 그 概要를 소개하고자 한다.

1 LP에 關聯된 Dimensionality 問題

LP (Linear Programming) 問題의 標準型(Canonical form)은 다음과 같이 주어지며

Minimize $\underline{c}^T \underline{x}$ (1)

subject to $A\underline{x}=\underline{b}$, $\underline{x} \geqq 0$ (단 A : $m \times n$ matrix)

이의 解는 一般的으로 Simplex Method[1),2)]에 의하여 구해질 수 있으나, Simplex Method 를 implementation 하는 過程에서, 選擇된 Basis Matrix B에 대한 dual cost vector (일명 shadow cost vector) 를 구하여야 하며 따라서 다음 方程式의 解를 求해야 한다.

$$B^T \underline{y} = \underline{c} \quad (2)$$

여기서 Matrix B는 basis vector 로써 構成된 ($m \times m$) Matrix 이며 vector $\underline{c}$ 는 basis 에 따라 vector $\underline{c}$ 의 element 로 구성되는 m차원 vector 이다.

式(1)에 의하여 주어지는 LP問題의 Matrix A는 일반적으로 sparse 하므로 非零要素의 一次配列 등에 의하여 記憶容量을 쉽게 줄일 수 있다. 式(2)의 解를 구하기 위해서는 Matrix B의 逆行列과 同等한 技能을 갖는 三角因數表(Triangular Factor Table)을 使用하며 計算時間의 短縮과 sparsity 를 最大限 保存함으로써 Memory 의 節減도 期할 수 있다. 하나 Simplex Algorithm 에서는 basis 를 change 함으로써 순차적으로 最適解를 찾기 때문에 每 Basis change 때마다 새로운 Basis Matrix 에 대하여 Factor Table 을 재구성하거나 또는 원래의 Factor Table 을 새로운 Basis Marix B에 對하여 修正해야 한다.[4)]

* 正会員 : 延世大 工大 電気工学科 助教授・工博

② B-G Decomposition에 의한 Basis Matrix의 Factor Table 修正 Algorithm[1),4)]

Simplex Algorithm에서 Basis Change의 경우 새로운 Basis Matrix에 대한 Factor Table의 再構成에는 計算時間이 많이 所要되므로 Bartels, Golub 等은 새로운 Basis Matrix를 Decomposition 함으로써 원래의 Factor Table을 修正하는 Algorithm을 開發하였다.

Basis Matrix B의 k 번째 column ($\underline{a}$라고 하자) 대신에 nonbasis vector $\underline{a}$를 basis로 取하는 경우를 고려해 보자. 그러면 새로운 Basis Matrix B는 다음과 같이 쓸 수 있다.

$$B = [B + (\underline{a} - \underline{a})\, e_k^T]\, P \qquad (3)$$

단 $P = [\underline{e}_1, \cdots\cdots, \underline{e}_{k-1}, \underline{e}_{k+1}, \cdots\cdots, \underline{e}_m, \underline{e}_k]$: Permutation Matrix

$\underline{e}_k = k$번째 elment 가 1인 unity vector

원래의 Basis matrix B는 $B = LU$의 관계가 있으므로 U를 다음과 같이 partition할 수 있다.

$$U = [U_1 \; L^1 \underline{a} \;\; U_2] \qquad (4)$$

(단 U_1, U_2는 각각 $k-1$개 및 $m-1$개의 column을 갖고 있음)

따라서 B는 다음과 같이 쓸 수 있다.

$$B = LH$$

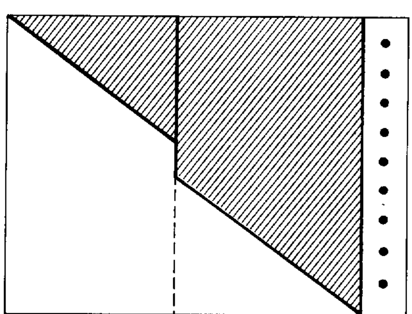
$H = [U_1, U_2 L^{-1}\underline{a}]$의 upper Heisenberg Matrix 형태를 갖는다.

지금 목적은 $B = LU$를 만족하는 三角行列 L, U를 찾는 것이므로 다음을 만족하는 Matrix M을 구하는 문제에 帰着되며

$$MH = U \qquad (5)$$

$$LM = L \qquad (6)$$

Matrix M은 다음같이 주어진다.



column k

그림 1.

$$M = M_{m-1}\, P_{m-1}\, M_{m-2}\, P_{m-2} \cdots\cdots M_k\, P_k \qquad (7)$$

단 $P_j = \begin{bmatrix} 1 & & & & 0 \\ & 1 \ddots & & & \\ & & 0 \;\; 1 & & \\ & & 1 \;\; 0 & & \\ & & & 1 \ddots & \\ 0 & & & & 1 \end{bmatrix}$ ……j 번째 row

$M_j = \begin{bmatrix} 1 & & & 0 \\ & 1 \ddots & & \\ & & 1 & \\ & & m_j\,1 \ddots & \\ 0 & & & 1 \end{bmatrix}$ ……j 번째 row

Base change에 따르는 새로운 Basis Matrix의 Factor Table을 구성하기 위하여 式(4)에서와 같이 B를 Decomposition하고 式(5),(6),(7)에 의하여 L,U를 update할 수 있다.

한편 Matrix A의 dimensionality가 크기 때문에 Matrix A와 Factor Table을 동시에 貯藏시킨다면 二重으로 memory가 所要된다. 따라서, Factor Table과 Matrix A의 Nonbasis 分部만을 Computer에 記憶시켜 두고 Matrix A의 basis vector가 필요한 경우는 Factor Table로부터 計算하는 方法을 取함으로써 Memory節減을 期한다.

結果的으로 Matrix A는 $A = [LU\; \underline{a}_{m+1} \cdots\cdots \underline{a}_n]$으로 Decomposition되어 저장되며 이 方法을 提案者의 이름을 따서 Bartels-Golub Decomposition Method라고 한다.

③ Factor Table에 의한 Base change와 dual 및 reduced cost vector의 updating[1),4)]

Simplex Method에서는 最適化 過程에서 Basis change가 必須的이며 Matrix A의 basis部分은 三角行列로 分解되어 있으므로 cost updating을 위하여 Simplex Tableau의 k번째 row 즉 pivot row를 計算하여야 한다.

Simplex Tableau의 k번째 row가 pivot row로 選定되었다고 하고 이 row vector를 v^T라고 하면 v^T는 다음 式에 의하여 주어진다.

$$B^T \underline{v} = \underline{e}_k \qquad (8)$$

上記式의 $\underline{v}$는 修正하기 전에 구할 수 있으나 Forward 및 Backword Elimination을 통하여 구하여야 하므로 計算時間을 增加시킨다. 다음의 Vector $\underline{v}$ 計算法은 修正된 Factor Table을 使用함으로써 단 한번의 Backward Elimimation을 통하여 $\underline{v}$를 구할 수 있다.

Basis change 의한 새로운 Basis Matrix B의 逆行列은, 式(3)에 Sherman-Morrison formula, 및 $B^{-1}\underline{a} = \underline{e}_k$, $P^T \underline{e}_k = \underline{e}_m$, $P^{-1} = P^T$의 關係를 이용함으로써 다음의 結果를 얻는다.

$$B^{-1} = [\{B + (\underline{a} - \underline{a})\, \underline{e}_k^T\} P]^{-1}$$

$$= P^T B^{-1} - \frac{(P^T B^{-1}\ \underline{a} - \underline{e}_m)\, \underline{e}_k^T\ B^{-1}}{\underline{e}_k^T\ B^{-1}\, a} \tag{9}$$

where $P = [\ \underline{e}_1, \cdots\cdots, \underline{e}_{k-1}\ ,\ \underline{e}_{k+1} \cdots\cdots \underline{e}_m\ ,\ \underline{e}_k]$

式(9)의 양변에 transpose를 취한 후 $\underline{e}_m$을 곱하면 $\underline{v}$를 얻는다.

$$\underline{v} = B^{-T} \underline{e}_m$$

$$= B^{-T} P\, \underline{e}_m - B^{-T}\ \frac{\underline{e}_k(\underline{a}^T\ B^{-T}\ P - \underline{e}_m^T)\ \underline{e}_m}{\underline{e}_k^T\ \ B^{-1}\,\underline{a}}$$

$$= B^{-T}\ \underline{e}_k\ /\ \underline{e}_k^T\ B^{-1}\,\underline{a} \tag{10}$$

따라서

$$\underline{v} = \alpha\ \underline{v} = \alpha\, B^{-T}\ \underline{e}_m = \alpha\ L^{-T}\ U^{-T} \underline{e}_m \tag{11}$$

$$\text{단}\quad \alpha = \underline{e}_k^T\ \ B^{-1}\ \underline{a} \tag{12}$$

여기서 α는 Pivot element 임에 유의하라.

式(11)에서 U는 Upper Triangular Matrix 이므로, $\underline{v}$는 다음과 같이 단지 한번의 Backward Elimination 통하여 구할 수 있다.

$$\underline{v} = \frac{\alpha}{U_{m,m}}\ \ L^{-T}\ \underline{e}_m \tag{13}$$

그리고, $\underline{v}$는 Factor Table 이 修正되고 난후에도 計算될 수 있음이 특기된다.

Section 1에 의하여 Factor Table이 修正될 수 있고 또 式(13)에 의하여 reduced Simplex Tableau 의 pivot row를 計算할 수 있으므로, basis change 에 따르는 모든 Data updating은 다음과 같이 할 수 있다.

(i) 원래의 basis에 대하여 Nonbasis variable 대한 reduced (or relative) cost 를 구한다.

$$z_j = c_j - \underline{\pi}^T\ \underline{a}_j \quad \text{for all nonbasirvectors } a_j \tag{14}$$

(ii) Dual (or Shadow) cost vector updating

$$\underline{\pi} = \underline{\pi} - Z_q\ \ \underline{v} \tag{15}$$

(iii) Reduced cost vector updating

$$z_j = \underline{\pi}^T\ \underline{a}_j = z_j\ - (\ \underline{v}^T\ \underline{a}_j\ )\ z_j \tag{16}$$

上記 updafing에서 dual cost vectcr π는 $B^T \underline{\pi} = c$의 관계로 부터 Factor Table을 使用하여 구할수도 있으나 式(15)에 의하여 updating하는 것이 훨씬 간편하다. 그리고 式(17)의 reduced cost vector updating에 있어서는, $\underline{v}$가 $\underline{\pi}$보다 더 sparse하므로, $z_j = \underline{\pi}^T\ \underline{a}_j$를 使用하는 것보다 $z_j - (\underline{v}^T\ \underline{a}_j\ ) z_q$를 使用하는 것이 計算時間을 적게 要한다는 것이 Tomlin에 의하여 알려졌다.

4 B-G Algorithm의 變種 (variants) [1)~3)]

(a) Saunder Algorithm[1)]

Saunder는 새로운 basis vector $\underline{a}$를 Matrix B의 마지막 column에 삽입하는 대신, 三角行例의 Spike 를 피할 수 있는 column에다 그림 2와 같이 삽입함으로써 Matrix M計算의 Permutation 회수를 줄이도록 하였다.

즉 그림 2의 Factor Table U의 scheme 을 얻기 위해서

$$P = [\,\underline{e}_{-1}\ \cdots\cdots \underline{e}_{-k-1},\ \underline{e}_{-k+1},\ \underline{e}_p,\ \underline{e}_k,\ \underline{e}_{p+1},\ \cdots\cdots,\ \underline{e}_m\,]$$

으로 하며 $P\underline{e}_p = \underline{e}_k$가 된다.

따라서 새로운 Basis Matrix $\hat{B}$에 대하여 Sherman — Morrison formular 를 적용시키면

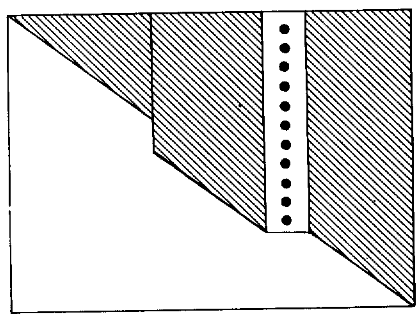
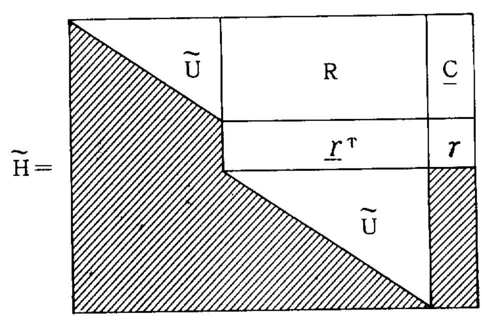


그림 2. Saunder algorithm의 U scheme



(사선친 부분의 element 는 Zero 임)

그림 3. Upper Heisenberg matrax scheme

$$\hat{B}^{-1} = PB^{-1} - \frac{(P^T B^{-1} \underline{a} - \underline{e}_p)\, \underline{e}_k^T B^{-1}}{\underline{e}_k^T B^{-1} \underline{a}}$$

그리고

$$\underline{v} = \alpha \; \underline{v} = \alpha B^{-T} \; \underline{e}_p$$

의 關係가 성립하며 $p-k$ row operation 을 하고난 후의 H matrix 는 그림 3의 upper Heisenberg 형태가 된다.

여기서, $\widetilde{H}$는 matrix H의 첫번째 P column 만을 포함하며, $p+1$번 이상의 column 들은 變化가 없으므로 考慮하지 않았다.

Saunder variants 에서는 pivot row 計算에 있어서 Factor Table 을 使用하여 $\hat{B}^T \underline{v} = \underline{e}_p$ 의 解를 구하는 것이 便利하며 다음 過程을 따른다.

$$\hat{U}^T \underline{w} = \underline{e}_p \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (19.a)$$

$$\hat{L}^T \underline{v} = \underline{w} \quad \cdots\cdots\cdots\cdots\cdots\cdots \quad (19.b)$$

式 (19.a)로 부터의 $\underline{w} = \hat{U}^{-T} \underline{e}_p$를 式(19.b)에 代入하면 $\underline{v} = \hat{L}^{-T} \underline{w}$로써 pivot row가 計算되며 여기서 $\underline{w}$는

$$\underline{w} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ r \\ -\frac{1}{r} \; U^{-T} \; \underline{r} \end{bmatrix} \qquad (20)$$

로 간단히 計算될 수 있다.

Saunder algorithm은 $\hat{U}^{-T} \underline{r}$의 計算을 要하지만 B-G algorithm과 比較하여 計算이 더많이 걸리는 경우는 없다는 것이 밝혀져 있다.

(b) Steepest Edge Simplex Algorithm [2),5)]

本 Algorithm은 B-G Decomposition 을 利用하여 Simplex Algorithm을 改善한 것으로써 Goldfarb 및 Reid 에 의하여 提案되었으며, 이들은 matrix A, cost vector $\underline{c}$ 및 solution vector $\underline{x}$ 를 basis및 nonbasic part 로 다음과 같이 partition한다.

$$A = [A_1 \;\; A_2] \qquad c = \begin{bmatrix} \underline{c}_1 \\ \underline{c}_2 \end{bmatrix} \qquad \underline{x} = \begin{bmatrix} A_1^{-1}b \\ \underline{O} \end{bmatrix} \qquad (21)$$

그리고 Matrix N을

$$N = \begin{bmatrix} A_1 & A_2 \\ O & I \end{bmatrix} \qquad (22)$$

로 定義하면, N의 逆行列은

$$N_1^{-1} = \begin{bmatrix} A_1^{-1} & -A^{-1}A_2 \\ O & I \end{bmatrix} \qquad (23)$$

로서 주어진다.

Matrix A의 columm q (nonbasis vector 임)를 basis vector, column p 대신에 basis 에 포함시킨다면 Solution vector $\underline{x}$는

$$\underline{X} = P_{Pq} \; (X = \theta \underline{\eta}_q) \qquad (24)$$

단 $\eta_i = N^{-1} e_i$, $i = 1, \ldots\ldots p, q \ldots n$

P_{Pq}: row p, q의 Permutation matrix 에 의하여 update 되며 θ는 $\underline{x}_q = 0$가 되도록 選定된다.

새로운 basis 選択은 reduced cost z_i 가 가장 큰 nonbasis vector 를 択하는 것이 効果的이며 Reduced cost z_i는

$$z_i = \underline{c}^T \underline{\eta}_i \;,\; i > m \qquad (25)$$

로 부터 計算되고, 이것을 normalizing하면

$$z_i^N = c^T \underline{\eta}_i / \| \underline{\eta}_i \| \qquad (26)$$

로 주어진다. 여기서 $\underline{\eta}_i$ 의 Norm $\|\eta_i\|$의 計算을 nonbasis vector 各各에 대해서 行한다면 상당한 時間이 所要되나 다행히 Matrix N의 修正해서 附隨的으로 計算될 수 있다.

row permutation P_{pq}에 의하여 Basis change 를 行했을 때, 새로운 Matrix $\bar{N}$는

$$N = (N + e_q \; (e_q - e_q)^T) P_{pq} \qquad (27)$$

로 나타낼 수 있으며 Sherman-Morrison formula 및 $\underline{e}_q^T N^{-1} = \underline{e}_q^T$ 의 關係를 適用시킴으로써

$$P_{pq} \bar{N}^{-1} = N^{-1} - \frac{N^{-1} \underline{e}_q \; (\underline{e}_p - \underline{e}_q)^T N^{-1}}{1 + (\underline{e}_p - \underline{e}_q)^T N^{-1} \underline{e}_q}$$

$$= N^{-1} - \frac{N^{-1} \underline{e}_q \; (\underline{e}_p^T N^{-1} - \underline{e}_q^T)}{\underline{e}_p^T N^{-1} \underline{e}_q} \qquad (28)$$

를 얻으며, e_i ($i > m$)를 post multiplying 함으로써

$$P_{pq} \; \bar{\eta}_q = \underline{\eta}_q / \underline{e}_p^T \; \underline{\eta}_q \qquad (29.a)$$

$$P_{pq} \underline{\eta}_i = \underline{\eta}_i - \underline{\eta}_p \; (e_p^T \underline{\eta}_i / e_p^T \underline{\eta}_q) (단 \; i \neq q) \qquad (29.b)$$

의 關係가 求해진다.

여기서 scalar α_i 및 $\bar{\alpha}_i$를

$$\alpha_i = -\underline{e}_p^T \; \underline{\eta}_i \;\;,\;\; \bar{\alpha}_i = \alpha_i / \alpha_q \qquad (30)$$

으로 定義하면 式 (29.a) 및 (29.b)는 다음과 같이 簡略化된다.

$$P_{pq} \underline{\eta}_q = -\underline{\eta}_q / \alpha_q \qquad (30.a)$$

$$P_{pq} \underline{\eta}_i = \underline{\eta}_i - \underline{\eta}_q \; \alpha_i \;,\; i \neq q \qquad (30.b)$$

上記 두 式으로 부터 目的하는 바의 updating formular를 다음과 같이 구할 수 있다.

$$\bar{r}_q = \bar{\eta}_q^T \bar{\eta}_q = r_q / \alpha_q^2 \qquad (31.a)$$

$$\bar{r}_i = \bar{\eta}_i^T \bar{\eta}_i = r_i - 2\bar{\alpha}_i \; \underline{\eta}_i^T \underline{\eta}_q + \bar{\alpha}_i^2 \alpha_q \qquad (31.b)$$

$$단 \quad r_k = \| \underline{\eta}_k \|^2 = \underline{\eta}_k^T \; \underline{\eta}_k \qquad (32)$$

上記 式에도 $\underline{\eta}_i = N^{-1} \underline{e}_i$의 計算이 包函되어 있으며 이의 計算에는 많은 時間이 所要되므로, 實際에

있어서는 다음의 Recurrence formula를 採択한다.

$$r_q = (1 + \| A_l^{-1} \underline{a}_q \|^2), \quad \bar{r}_q = r_q / \alpha_q^2$$

$$\bar{r}_i = r_i - 2\bar{\alpha}_i a_i^T A_l^{-1} \underline{a}_q + \bar{\alpha}_i^2 r_q \quad (33.b)$$

위의 式은 $\underline{\eta}_i$의 定義 (즉 $\underline{\eta}_i = N^{-1} e_i$)로 부터 α_i 및 α_q가 각각 $A_l^{-1} A_2 \underline{e}_i$ 및 $\bar{A}_l^{-1} \bar{A}_2 \underline{e}_i$의 P번째 component 임을 유의하면 쉽게 證明될 수 있다.

한편 reduced cost vector Z는 다음 式에 의하여 간단히 update 된다.

$$\bar{z}_q = \underline{\bar{c}}^T \bar{\eta}_q = \underline{c}^T P_{pq} \bar{\eta}_q = -z_q / \alpha_q \quad (34.a)$$

$$\bar{z}_i = \underline{\bar{c}}^T \underline{\bar{\eta}}_i = \underline{c}^T P_{pq} \underline{\bar{\eta}}_i = z_i - z_q \bar{\alpha}_i, \; i \neq q \quad (34.b)$$

結論的으로, 上記 過程을 따라 basis change를 式(6)의 $z_i^{*} (= z_i / r_i)$가 最大가 되도록 하면 steepest edge를 따라 最適化하는 것이 되며, updating에 의한 反復計算에 의하여 最適解를 求할 수 있다.

本 Algorithm의 全 過程을 要約하면 다음과 같다.

(a) Optimality condition을 check하라.

만약 조건 $z_i \geq 0 \; (i = m+1, \cdots\cdots n)$ 이 성립하면, 현재의 解를 最適解로 出力하고 計算을 終了하며 그렇지 않으면 step(b)로 가라.

(b) pivot column i를 다음에 의하여 選択하고

$$\text{Maximize} \left\{ \frac{z_i^2}{r_i} \middle| Z_i < 0 \right\} = \frac{z_q^2}{r_q} \quad (35)$$

Factor table에 使用하여 pivot column을 計算하라.

$$\underline{w} = A_l^{-1} \underline{a}_q = U_l^{-1} L_l^{-1} \underline{a}_q \quad (36)$$

(c) problem의 boundness를 check하라.

만약 모든 w_i ($i = 1, \cdots, m$)가 0보다 적거나 같으면 unbound case 임으로 計算을 中斷하고 그렇지 않으면 다음 step으로 가라.

(d) Pivot Element 를 다음에 따라 決定하고

$$\underset{1 \leq i \leq m}{\text{Minimize}} \left\{ \frac{x_i}{w_i} \middle| w_i > 0 \right\} = \frac{x_p}{w_p} \quad (37)$$

Pivot element w_p를 별도로 記憶시킨다.

(e) 解를 修正하라

$$\bar{x}_p = x_p / w_p \quad (38)$$

$$\bar{x}_i = x_i - w_i \bar{x}_p, \; i = 1, 2, \cdots, m, \; i \neq p$$

(f) Z, r 및 A_l^{-1}의 Factor Table 修正하라.

$$\bar{z}_q = \underline{c}^T \underline{e}_q - c_1^T \underline{w}$$

$$\bar{r}_q = 1 + \underline{w}^T \underline{w} \quad (39)$$

$$\bar{w} = A_l^{-T} \underline{w}$$

• B-G Algorithm에 의하여 $\bar{A}_l$에 對한 Factor Table을 構成하라.

• Dual cost를 $\underline{y} = \bar{A}^{-T} \underline{e}_p$에 의하여 計算하라.

• z, r를 修正하라 (단, $i = m+1, \cdots n, \; i \neq q$)

$$\bar{a}_i = \underline{y}^T \bar{a}_i \quad (40)$$

$$\bar{z}_i = z_i - \bar{a}_i z_q \quad (41)$$

$$\bar{r}_i = max\{ (r_i - \bar{2}\alpha_i \underline{w}^T \underline{a}_i + \bar{a}_i^2 r_q) \; (1 + \bar{\alpha}_i^2) \} \quad (42)$$

$$\left. \begin{aligned} \bar{z}_q &= -z_q / w_p \\ \bar{r}_q &= r_q / w_p^2 \end{aligned} \right\} \quad (43)$$

이 外에도 差異點은 近小하지만 Harris, Crowder - Hattingh, Dentzig, Reid 等에 의하여 提案된 多数의 変種 (variants)이 있다.

5 B-G Algorithm의 應用事例 및 檢討

以上에서 紹介한 바와 같이 B-G decomposition 方法은 Factor Table을 使用함과 同時에 basis change에 對한 그의 修正을 容易하게 함으로써 計算時間과 所要記憶容量을 대폭 줄일 수 있으므로 LP의 適用範圍를 크게 擴長시켰다.

B-G aigorithm의 適用例로는, Chan-Yip 等이 steepest edge algorithm을 LSGR (Load Shedding and Generator Rescheduling) 問題에 適用을 試圖한 바 있다.[5] LSGR 問題에서는 発電費用이 非線形函数로 주어지므로 LP 適用을 위해서는 piecewise linearization을 通하여 LP 問題로 変形시켜야 하며, 이 경우 piecewise linearization에 따른 不等式 및 Slack variable이 追加됨에 따라 Dimensionality 問題를 大幅 加重시키므로 從來의 LP 技法 適用은 計算時間과 computer memory size 面에서 거의 實用性이 없는 것으로 생각되었으나, Chan과 Yip은 그들의 硏究를 通하여 B-G decomposition에 의한 LP algorithm의 LSGR 問題에 대한 實用可能性을 立證하였다.

뿐만 아니라 Large Dimensionality와 關聯된 工學問題에 있어서 system matrix가 몇개의 element만 変更되는 경우는 흔히 볼 수 있으며, 이들의 inverse matrix와 vector와의 演算이 큰 問題가 되는 경우가 많으며 이 경우에 Matrix column 変化에 對한 Factor Table 修正이 容易한 B-G decomposition algorithm을 適用시킬 수 있다. [6]

그 例로서 電力系統에 있어서의 故障電流計算, 想定事故解析 (Contingency Analysis) 等에 直接 適用시킬 수 있으며, 最適潮流 (Optimal Load Flow) 計算等의 非線最適化問題도 piecewise linearization

을 통하여 B－G decomposition에 의한 LP algorithm을 適用시킬 수 있다.〔6〕특히 最適潮流計算問題에 있어서는 目的函数가 非線形일 뿐만 아니라 많은 不等式制約條件이 包函되어 있으며, 이 問題에 대해서는 일반적으로 NLP (Nonlinear Progrmming) 技法을 使用해 왔으나, Algorithm이 発散하다든지 혹은 収検하는 경우에도 計算時間이 많이 걸리는 等의 問題點이 指摘되어 왔다. 하지만 piecewise linearization을 통하여 이 問題에 B－G Algorithm을 適用시킨다면 計算時間의 短縮을 期할 수 있을 뿐만 아니라 LP algorithm의 性質上 stable한 solution을 保障받을 수 있는 利點도 있다.

또한 Quadratic Optimization 問題는 Wolfe's algorithm에 의하여 LP問題로 바뀌어 질 수 있으며 여기에 B－G algorithm을 適用시킬 수 있다.

参 考 文 献

1) D. Goldfarb, "On the Bartels-Golub decomposition for linear programming basis", Report No. CSS 18, Harwell, U.K., Automatic Energy Research Establishment, August 1975.

2) D. Goldfab and J.K. Reid, "A practicable steepest edge simplex algorithm", Report No. CSS 19, ibid., May 1975.

3) J.K. Reid, "A Sparsity-Exploiting Variant of the Bartels-Golub decomposition for Linear Programming Bases", Report No. 20, ibid., August 1975.

4) J. Franklin, Methods of Mathematical Economics., Spring-Verlag, 1980.

5) S.M. Chan and E. Yip, "A solution of the transmission limited dispatch problem by sparse linear programming", IEEE Trans., Power Apparatus and Systems, vol. PAS-98, No. 3, pp.1044—1055 May/June 1979.

6) Y.H. Moon, "Optimal Load Shedding and Generator Rescheduling for Overload Suppression in Large Power Systems", Ph.D Thesis, Oregon State Univ., pp.73—164, Jun 1983.