

数值解析에 의한 最適化 設計 技法의 比較 研究 (Comparative Studies on Numerical Optimal Design Techniques)

趙 宣 彙*· 朴 鍾 根**
S. H. Cho, J. K. Park

Abstract

Computer codes on two numerical optimization methods—Sequentially Unconstrained Minimization Technique (SUMT) and Gradient Projection Method—are constructed and tested with several test problems. Design formulation of tension - compression coil spring is set up and the solution is obtained. Consequently, the feature, the advantage and the limitation of these methods, made clear through the tests, are discussed.

1. 序 論

最近 有限次元의 문제에 있어서 数值·解析에 의한 非線型 最適化 技法이 여러學者들에 의해 研究되고 있다.⁽¹⁾⁽²⁾⁽³⁾⁽⁴⁾ 그 중 重要한 技法들—SUMT (sequentially unconstrained minimization technique) 그리고 勾配投像法 (gradient projection method) —에 대해서 computer code 를 作成해서 몇가지 test problem 으로 그 收斂特成 및 각각의 特徵과 長短點을 알아보고 構體的인 設計問題에 이 code 를 應用해 보았다.

數理計劃法의 最適設計에 関한 式은 아래와 같이 要約된다. 즉 目的函數 $f(x)$ 를 最小化하는 設計變數 x 를 決定함에 있어서 x 가 k 次 矢벡터 空間上에 있는 點일 때, 이 設計變數 x 는 다음과 같은 n 개의 等式 拘束條件과 $(m-n)$ 개의 不等式 拘束條件을 만족시켜야 한다.

$$\phi_i(x) = 0, \quad i=1, \dots, n \dots\dots\dots(1)$$

$$\phi_i(x) \leq 0, \quad i=n+1, \dots, m \dots\dots\dots(2)$$

即, 式(1)은 等式拘束條件의 境遇이고, 式(2)가 不等式 拘束條件을 表示한다.

2. 理論的 考察

2.1 SUMT

SUMT 이론에서는 目的函數 $f(x)$ 에 罰則項 (penalty term) 을 합한 罰則函數의 最小點을 구함으로써 拘束條件 있는 最小化 問題를 拘束條件 없는 最小化 問題로 轉換시킨다. 이때 罰則函數는 아래와 같이 정의된다.

$$P(x, \lambda_1, \lambda_2) = f(x) + \lambda_1 \left[\sum_{i=1}^n \left(-\frac{1}{\phi_i} \right) \right] + \lambda_2 \left[\sum_{i=1}^n \phi_i^2 \right] \dots\dots\dots(3)$$

式(3)에서는 等式 拘束條件과 不等式 拘束條件이 동시에 使用되고 있으므로 混入型 (mixed type)⁽⁵⁾ 이다. 本 code 에서 λ_1, λ_2 를 아래와 같이 變化시켰다.

$$\lambda_1' = (0.7071)' \times 1.0, \quad \lambda_2' = (1.4142)' \times 0.1;$$

$$\text{反復回數 } j = 0, 1, 2 \dots\dots\dots(4)$$

λ_1, λ_2 값이 式(4)와 같이 變化되는 罰則函數 f (3)

* 正會員 서울大 工大

** 正會員 慶南大 工大

의 最小點을 구하는 拘束條件 없는 最小化 技法으로 conjugate gradient algorithm⁽⁵⁾⁽⁶⁾⁽⁷⁾을 이용했다. 이 알고리즘에서 探索方向(search direction) S는 아래와 같이 計算된다.

$$S^{j+1} = -\nabla P(x^{j+1}, \lambda_1, \lambda_2) + \left[\frac{\|\nabla P(x^{j+1}, \lambda_1, \lambda_2)\|^2}{\|\nabla P(x^j, \lambda_1, \lambda_2)\|^2} \right] S^j \dots (5)$$

이 때 搜索方向의 安定을 위하여 k회 反復 때마다 搜索方向을 遞減方向(steepest descent direction) $-\nabla P(x, \lambda_1, \lambda_2)$ 으로 조정해 준다.⁽⁵⁾⁽⁶⁾

Fig. 1. 및 2.는 SUMT의 흐름圖를 表示한다. 여기서 $\delta(x)$ 는 다음과 같이 계산했으며, 主 프로그램에서는 0.001 이하로 될 때까지, 그리고 conjugate gradient algorithm에서는 $(0.9)^j \times 0.001$ (j 는 主프로그램의 反復回數) 이하로 될 때까지 計算을 遂行하였다.

$$\delta(x) = \sqrt{\left[\sum_{i=1}^k \left(\frac{x_i^{j+1} - x_i^j}{x_i^j} \right)^2 \right] / k} \dots (6)$$

또한 $P(x + \alpha S, \lambda_1, \lambda_2)$ 를 最小化하는 스텝라 α 를 구하는 방법으로는 速性分割探索 技法(golden section search technique)⁽⁵⁾⁽⁶⁾⁽⁷⁾을 利用하였다. 罰則函數의 勾配는 아래와 같이 計算하였다.

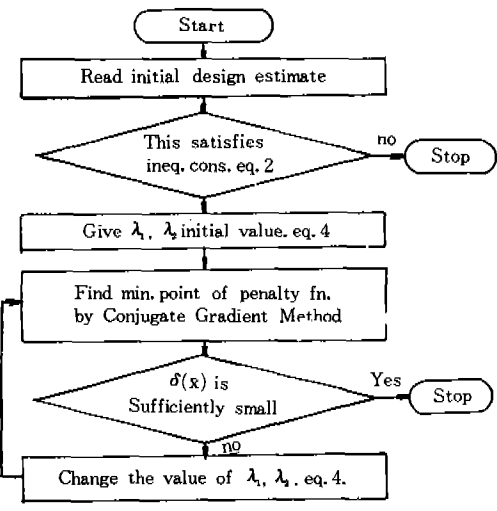


Fig. 1. Flow chart of main program.

$$g_i'(x) = \lim_{\Delta x_i \rightarrow 0} \left\{ [P(x_1, \dots, x_i + \Delta x_i, \dots, x_k, \lambda_1, \lambda_2) - P(x, \lambda_1, \lambda_2)] / \Delta x_i \right\};$$

$$i = 1, \dots, k \dots (7)$$

$$g_i''(x) = \lim_{\Delta x_i \rightarrow 0} \{ [P(x, \lambda_1, \lambda_2) - P(x_1, \dots, x_i - \Delta x_i, \dots, x_k, \lambda_1, \lambda_2)] / \Delta x_i \};$$

$$i = 1, \dots, k \dots (7)'$$

$$g_i = \frac{1}{2} (g_i' + g_i''); \quad i = 1, \dots, k \dots (7)''$$

여기서,

$$\Delta x_j' = -0.01 (0.8)^j |x_j|; \quad j = 0, 1, \dots (7)'''$$

이때 $|(g_i' - g_i'') / g_i''| \leq 0.01$ 이 되고 또한 $|(g_i^{j+1} - g_i^j) / g_i^j| \leq 0.01$ 이 될때의 g_i 의 값을 勾配로 취하였다. 이와같은 勾配計算方法을 다음의 勾配投像法에서도 利用하였다.

2.2 勾配投像法⁽⁵⁾⁽⁶⁾

이 技法의 基本原理는 目的函數 $f(x)$ 의 遞減方向 $-\nabla f(x)$ 를 投像行列 P에 投像시키면 그 影像의 方向은 可用區域(feasible region)의 境界

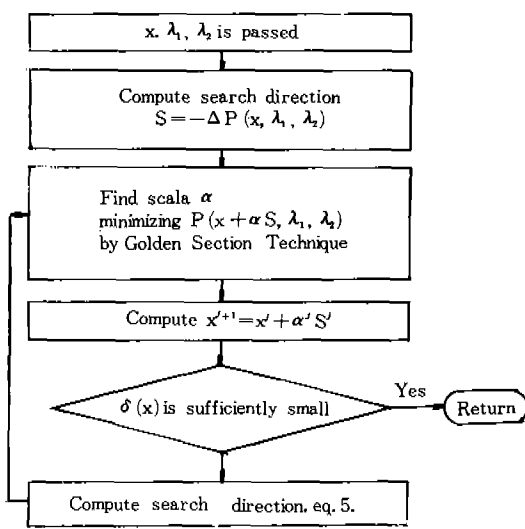


Fig. 2. Flow chart of Conjugate Gradient Method

에 접하며 이 방향으로 設計變數를 移動 시키는 것이다. 즉,

$$g = \nabla f_{(x)} = \left(\frac{\partial f}{\partial x_1} \quad \frac{\partial f}{\partial x_2} \quad \dots \quad \frac{\partial f}{\partial x_k} \right)^T \dots (8)$$

$$G = [\nabla \phi_1 \quad \nabla \phi_2 \quad \dots \quad \nabla \phi_l]$$

$$\begin{bmatrix} \frac{\partial \phi_1}{\partial x_1} & \frac{\partial \phi_2}{\partial x_1} & \dots & \frac{\partial \phi_l}{\partial x_1} \\ \frac{\partial \phi_1}{\partial x_2} & \frac{\partial \phi_2}{\partial x_2} & \dots & \frac{\partial \phi_l}{\partial x_2} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \phi_1}{\partial x_k} & \frac{\partial \phi_2}{\partial x_k} & \dots & \frac{\partial \phi_l}{\partial x_k} \end{bmatrix}$$

여기서 l 은 等式 拘束條件의 個數 n 에 ϵ -敏感 位置를 $(\phi_i(x) \geq -\epsilon; i = n+1, \dots, m; \epsilon$ 은 작은 陽數) 考慮한 不等式 拘束條件의 個數를 合한 數로서 設計變數의 個數 k 보다 적어야 한다. 또한,

$$d = [-\phi_1(x) \quad -\phi_2(x) \quad \dots \quad -\phi_l(x)]^T \dots (10)$$

이때 投像行列 P 는 다음과 같이 計算된다.

$$P = I - G(G^T W^{-1} G)^{-1} G^T W^{-1} \dots (11)$$

여기서 W 는 設計變數評價行列로서 設計變數의 값이 서로 크게 다를 수 있는 工學的 計算에서 필요하며 本 코오드에서는 아래와 같이 計算하였다.

$$W = \begin{bmatrix} 1/x_1^2 & & & 0 \\ & 1/x_2^2 & & \\ 0 & & \dots & 1/x_k^2 \end{bmatrix} \dots (12)$$

反復施行範圍의 크기(step size)를 아래와 같이 計算하여 全 反復計算을 通하여 一定하게 두었다.

$$\gamma = \frac{f_{(x)} (\Delta f_{(x)} / f_{(x)})}{g^T W^{-1} P g} \dots (13)$$

$(\Delta f_{(x)} / f_{(x)})$ 는 첫번째 反復計算에서 目的函數의 減少率로서 本 코오드에서 最適點으로 갈수록 기울기가 완만해지는 경우에는 -20% 전후로, 그리고 勾配가 急해지는 境遇에는 -10% 前後로

잡았다(目的函數가 陰數일 때는 +부호가 됨).

Kuhn-Tucker 의 必要條件⁽⁵⁾⁽⁶⁾⁽⁷⁾에서의 乘數 v 는 아래와 같이 計算된다.

$$v = -(G^T W^{-1} G)^{-1} \left[G^T W^{-1} g + \frac{1}{r} d \right] \dots (14)$$

여기서 計算된 v 의 成分중 陰數가 있을 경우 그것에 해당되는 不等式 拘束條件을 無視하고 (式 (7), (8) 에서) 다시 計算하면 더 나은 點으로 移動이 可能하게 된다.

$$\delta x^1 = \gamma W^{-1} P (-g) \dots (15)$$

$$\delta x^2 = W^{-1} G (G^T W^{-1} G)^{-1} d \dots (16)$$

여기서 δx^1 은 遞減方向 $-g$ 를 P 에 投像한 影像이며 δx^2 는 設計變數가 可用區域에서 멀어졌을 때 보정해주는 役割을 한다. (Fig. 3 참조)

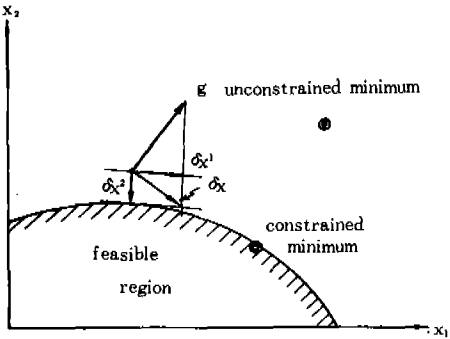


Fig. 3. Direction of $\delta x, \delta x^1, \delta x^2$

$$x^{i+1} = x^i + \delta x^i = x^i + \delta x^{i1} + \delta x^{i2} \dots (17)$$

式(6)과 같이 定義한 $\delta(x)$ 가 0.001 이하가 되고 設計變數의 값이 0.001의 誤差限界 내에서 拘束條件을 滿足시킬 때 計算을 끝내게 하였다.

이 勾配投像法에서는 等式拘束條件이 없고 ϵ -敏感值를 考慮한 不等式 拘束條件이 없는 點을 設計變數가 通過할 때는 移動이 不可能하게 된다.

따라서 이런 境遇 拘束條件없는 最小化問題로 생각할 수 있으므로 $(P=I)$ 아래와 같이 코오드를 作成하여 設計變數를 移動시켰다.

$$\delta x = \{ f_{(x)} (\Delta f_{(x)} / f_{(x)}) / (g^T W^{-1} g) \} W^{-1} g ;$$

$$x^{i+1} = x^i + \delta x^i \dots (18)$$

3. 計算의 實行 및 그 結果

여기서 遂行한 計算問題는 프로그램 코오드의 比較와 結果의 타당성을 確認하기 爲하여 이미 提示된 問題^{2), (5)}를 擇하였다.

prob. 1. $f_{(x)} = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1) / 10^3$

拘束條件; $0 \leq x_i \leq 10, i = 1, \dots, 4$

prob. 2. $f_{(x)} = -(0.44x_1^2x_2^{-2} + 10x_1^{-1} + 0.592x_1x_2^{-3})$

拘束條件; $1 - 8.62x_1^{-1}x_2^3 \leq 0, 1 \leq x_1 \leq 5, 0.5 \leq x_2 \leq 5$

prob. 3. $f_{(x)} = -0.0201x_1^4x_2x_3^2/10^6$

拘束條件; $0 \leq x_1 \leq 36, 0 \leq x_2 \leq 5, 0 \leq x_3 \leq 125, 675 - x_1^2x_2 \geq 0, 0.419 - (x_1, x_3)^2/10^7 \geq 0$

prob. 4. 〈引張-壓縮코일 스프링〉

$$f_{(x)} = \frac{\pi^2(N+2)Dd^3\rho g}{4} \text{ (무게)}$$

拘束條件; $K = \frac{d^4G}{8D^3N} = 2500,$

$$\frac{8DP}{\pi d^3} \left[\frac{4D-d}{4D-4d} + \frac{0.615d}{D} \right] \leq \tau_a$$

$$= 4 \times 10^6, \text{ 여기서 } P = 10,$$

$$10 = f_1 \leq \frac{d}{2\pi D^2 N} \sqrt{\frac{G}{2\rho}} \leq f_2 = 100,$$

$$D + d \leq 0.15, d \geq 0, 2d \leq D,$$

$$N \geq 5, 6000 \leq \rho \leq 10000,$$

$$2 \times 10^{10} \leq G \leq 6 \times 10^{10}.$$

여기서의 단위는 전부 SI단위이며, N(有效卷數), D(스프링直徑), d(coil直徑), ρ(밀도), g(重力加速度), K(스프링常數), G(剪斷彈性係數), f_1, f_2 (最小 및 最大振動數), P(最大荷重), τ_a (許容 剪斷應力), 設計變數 $x = \langle NDd\rho G \rangle$.

특히 prob. 4의 境遇에 있어서는 設計變數의 次元(dimension)을 높이기 爲해서 密度ρ 및 剪斷彈性係數 G가 追加되었으며 스프링常數 K를 等式 拘束條件으로 取扱하였다.

SUMT에 의한 計算 結果를 Table 1., 勾配投像法에 의한 計算 結果를 Table 2에 실었다. 또한 反復回數에 따른 目的函數의 變化를 Fig. 4, Fig. 5에 실었다.

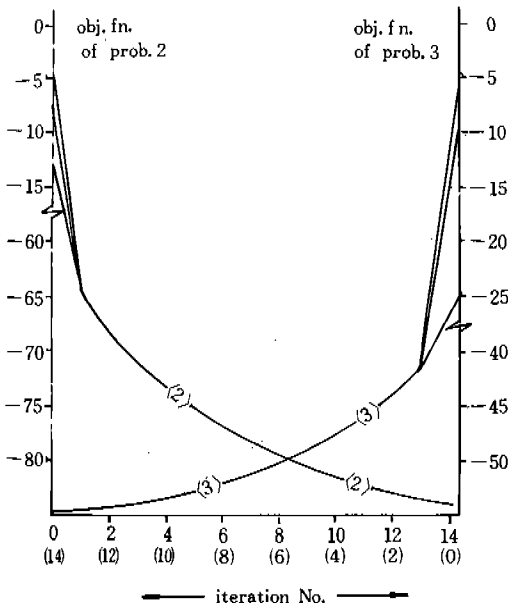


Fig. 4. Change of obj. fn subject to iteration No. by SUMT

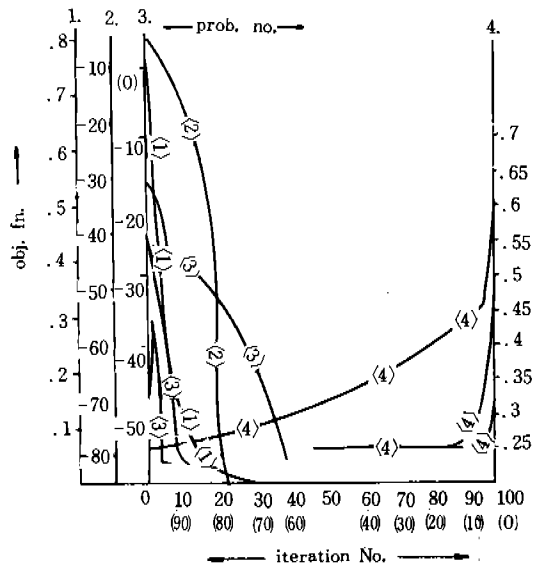


Fig. 5. Change of Obj. fn subject to iteration No. by G. P. M.

Table 1. Output by SUMT

Prob. No.	initial point	optimal point	obj. fn.	iter. No.	CPU time
1	5, 5, 5, 5	1.003 1.005 1.004 1.007	1.473×10^{-6}	19	2 M 13S 125MS
2	2.5 2.5	4.966 0.8354	-84.27	15	5 M 44S 418MS
	1.5 0.7	4.956 0.8356	-83.76	14	
	4 4	4.957 0.8353	-83.88	14	
3	20 1.2 80	24.97 1.065 81.31	-54.97	13	7 M 33S 715MS
	10 4 100	21.54 1.431 94.25	-54.96	13	
	30 0.5 30	30.90 0.6952 65.69	-54.97	13	

<Computer ; FACOM BOS/VS>

Table 2. Output by G. P. M.

prob. No.	initial point	optimal point	obj. fn.	iter. No.	$\Delta f/f$	CPU Time
1	2 2 2 2	1.012 1.026 1.014 1.028	2.974×10^{-5}	71	-0.2	46S 725MS
2	2.5 2.5	5.0 0.8384	85.28	22	0.1	22S 557MS
3	22.3 0.5 80	31.84 0.6659 64.29	-56.85	41	0.07	1 M13S 135MS
	20 1.2 80	23.34 1.239 87.69	-56.85	10		
	20 1 125	23.14 1.261 88.46	-56.85	5		
4	18 0.085 0.0086	14.80 0.06297 0.007799	0.2507	17	-0.1	2 M55S 608MS
	6000 3.8×10^{10}	6000 2.0×10^{10}				
	17 0.068 0.0079	14.76 0.06306 0.007799	0.2507	52		
	6000 2.6×10^{10}	6000 2.0×10^{10}				
19 0.09 0.009	15.00 0.06250 0.007779	0.2508	100			
6000 3.9×10^{10}	6000 2.0×10^{10}					

<computer ; FACOM BOS/VS>

4. 結果의 考察

Prob. 1은 拘束條件이 있으나 最小點이 可用區域의 境界에 接하지 않으므로 拘束條件 없는 最小化 問題의 境遇이다. Prob. 3을 제외한 計算 結果의 最適點은 모두 서로 一致하고 있다. Prob. 3의 경우는 最適點이 여러곳에 있는 特殊한 境遇로 볼 수 있으며, Eason & Fenton의 考察²⁾에서도 最適點이 명시되어 있지않고 目的函數의 값만 提示되었다.

SUMT에 의한 目的函數의 變化는 反復回數에 따라 勾配가 완만해지지만(Fig. 4) GPM에 의한 變化는 目的函數에 따라 勾配의 變化가 正反對가 되었다(Fig. 5.). 즉, 最大點(陰의 最小值)을 구하는 경우에는 反復回數가 많아질수록 기울기가 급해지고 陽의 最小點을 구하는 경우에는 SUMT와 마찬가지로 變化가 생기는 것을 볼 수 있다. 이것은 目的函數의 勾配에 기인하는 것으로 볼 수 있다.

SUMT에 의한 계산에서 設計變數의 初期值를

서로 다르게 주더라도 目的函数的 값은 첫번째 반복계산에서부터 서로 일치하고 있다. 이것은 반복회수에 따라 λ_1, λ_2 의 값이 똑같이 변화하도록 놓았기 때문이다. 極小점이 여러곳에 존재하는 문제는 設計變數의 初期值를 다르게 하여 여러 번 계산해볼 필요가 있다. 왜냐하면 λ_1, λ_2 의 初期值는 罰則函数에서 罰則項의 영향을 증가 혹은 감소시키고 罰則項의 영향이 지나치게 클 때에는 設計變數의 初期值를 바꾸더라도 極小点으로만 수렴하는 경우도 있기 때문이다(Fig. 6.).

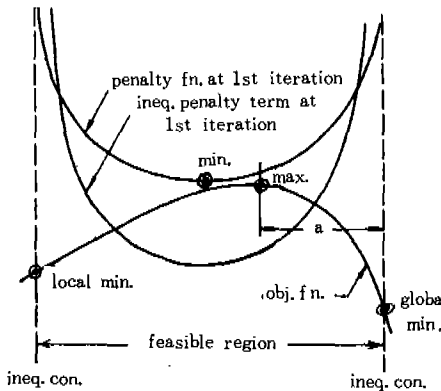


Fig. 6. Convergence to local min.

設計變數가 最小点으로 수렴할 수 있게하기 위해서는 최초의 반복계산에서 罰則函数的 極小점이 Fig. 6의 구역 a에 적어도 한 개 존재해야 한다.

SUMT의 장점은 논리가 단순하고 비교적 정확한 最適點을 구할 수 있는 반면 반복회수가 많아짐에 따라 罰則項이 깊은 골짜기를 형성하므로 設計變數의 이동이 곤란해질 때가 있다. 이런 단점을 보강하는 새로운 방법⁽⁶⁾이 제시되기도 한다.

勾配投像法에 의한 目的函数的의 변화에서 prob. 3과 같이 곡선이 꺾이는 부분은 設計變數가 이동하는 도중 새로운 ϵ -敏感值拘束條件을 만나서 방향이 급격하게 전환하는 경우이며 目的函数的의 값이 첫 반복계산에서 증가하는 것은 拘束條件의 誤差를 보정하는 δx^2 의 영향이다. GPM의 장점은 비교적 높은 非線型性에서도 수렴성

능이 우수하고 짧은 시간에 충분히 정확한 最適點을 구할 수 있으나 設計變數의 初期值에 따라서 반복회수가 많아지는 단점이 있다. 이런 점은 계산 도중 反復施行範圍의 크기를 變化시킴으로써 반복회수를 줄일 수 있을 것으로 생각된다.

5. 結論

(1) SUMT에 의한 最適點의 수렴여부는 λ_1, λ_2 의 初期值와 이의 반복회수에 따른 變化率과 관계가 깊다. 즉 λ_1, λ_2 를 급격하게 변화시키면 수렴속도는 빨라지지만 이동도중 깊은 골짜기에서 머무르는 경우가 생기고 λ_1, λ_2 를 서서히 변화시키면 수렴속도는 느리지만 좀더 확실히 最適點으로 접근시킬 수 있다.

(2) 極小점이 여러 곳에 존재하는 문제는 첫 번째 반복계산에서 罰則項의 영향을 적게하고, 設計變數의 初期值를 서로 다르게 잡아서 여러 곳에서 수렴시키는 것이 바람직하다.

(3) 勾配投像法에 의한 계산에서는 設計變數의 初期值에 따라 반복회수의 변화가 크다 (Fig. 5 참조).

(4) 계산 시간은 許容誤差에 크게 좌우되지만 許容誤差를 같게 했을 때 SUMT가 GPM보다 더 많은 시간을 소요한다(Tab. 1, Tab. 2 參照).

參 考 文 獻

- (1) G. A. Gabriele, K. M. Ragsdell "The Generalized Reduced Gradient Method; A Reliable Tool for Optimal Design," Journal of Engineering for Industry, Trans ASME, Vol. 99, No. 2, pp. 394-400, 1977.
- (2) E. D. Eason, R. G. Fenton, "A Comparison of Numerical Optimization Methods for Engineering Design," Journal of Engineering for Industry, Trans ASME, vol. 96, No. 1, pp. 196-200, Feb. 1974.
- (3) S. B. Schuldt, G. A. Gabriele, R. R. Root, E. Sandgren, K. M. Ragsdell, "Application of a

New Penalty Function Method to Design Optimization," Journal of Engineering for Industry, Trans ASME, vol. 99. No. 1, pp. 31~36, Feb. 1977.

- (4) J. N. Siddall, "Frontiers of Optimal Design," Journal of Mechanical Design, vol. 101, Trans ASME, pp. 674~681, Oct. 1979.
- (5) E. J. Hang, J. S. Arora ; "Applied Optimal Design," Wiley 1979.
- (6) Mordecai Avriel "Nonlinear Programming," Prentice Hall, 1976.
- (7) K. J. Arrow, L. Hurwicz, H. Uzawa, "Studies in Linear and Nonlinear Programming," Stanford University, 1972.