

活性的な制約面の選択を主眼にした最適化問題の解法

AN OPTIMIZATION METHOD USING SELECTION TECHNIQUES OF ACTIVE CONSTRAINT SURFACES

平田 恭久*・伊藤 文人**

By Yasuhisa HIRATA and Fumihito ITOH

There are a large number of inequality constraints in a structural optimum design problem, however, an optimum design point is usually located on active constraint surfaces that are equivalent to equality constraint. If we select such equality constraints among the inequality constraints, the inequality constrained problem can be transformed into an equality constrained problem. An optimum point of the new problem that is subjected to the selected equality constraints is obtained by using the Lagrange multiplier technique. In this paper an optimization method in which selection techniques are used is proposed, and a simple example on optimum solutions of composite plate girders is shown.

Keywords: optimization, girder, selection, constraint, method

1. ま え が き

設計問題ではなんらかの条件が課せられており、これらの条件は不等号条件の形で表わされている場合が多い。このような条件下で目的関数として選んだものの最小化を計ると、これは最適設計問題になる。一般の設計問題では最小化を明確に意識していなくても、ある条件のもとでよりよい設計結果を得ようとしている点では、最適設計問題と本質的に同じである。

設計問題には数多くの不等号条件があるが、個々の設計結果ではそのうちの一部が実際に効いている条件になるので、設計者は数多くの条件の中から、どの条件で設計結果が定まるかに関心をもたざるを得ない。ある不等号条件で設計結果が定まることは、その条件が等号条件として働いていることであり、これは解が等号条件として働いている制約面の上に乗っていることを意味している。

実際に効いている制約条件は許容領域に直接面した制約面であり、これを活性的な制約面とよぶ。活性的な制約面

であっても、ある制約条件を等号で満足していることが目的関数の改善に対し不利になることがあるので、このような不利なものを含まないことを強調するときは活性的でかつ有利な制約面とよぶ。不等号条件が等号条件として働いていることを示すため、以下ではこの状態を等式制約とよぶが、活性的でかつ有利な制約面になっていることは等式制約が具備すべき条件になる。

活性的でかつ有利な制約面よりも目的関数が改善される状態はないので、設計問題の解は基本的には活性的でかつ有利な制約面でなければならない。設計問題の解は最終的にはある一点を指定することになるが、比較検討の段階ではもっと広い範囲について解の状態を把握することが重要であり、これは活性的でかつ有利な制約面上を調べることになる。

一般の設計問題と最適設計問題との本質的な面での共通性からして、著者らは最適設計問題では実際に効いている制約条件を明らかにし、その制約面上で最適解を探索していくことが有効であり、このためには活性的でかつ有利な制約面上を探索できる方法が必要であると考えた。

この必要性に対し、従来からある非線形最適化手法が十分に対応できるものであるかを、最適構造設計で用いられる代表的な最適化手法である罰金関数法、逐次線形

* 正会員 群馬高専助教授 土木工学科
(〒371 前橋市鳥羽町 580)

** 正会員 工博 東京都立大学教授 工学部土木工学科
(〒158 世田谷区深沢 2-1-1)

計画法，許容方向法について調べると^{2)~5)}，これらの手法は次のように解釈できる。

罰関数法は制約面上に乗りにくい構造である。逐次線形計画法では線形近似しているため，本来の非線形制約面に乗っている保証はない。許容方向法では非線形制約面に抵触しないように探索方向を選ぶため，非線形制約面には乗っていないのが普通である。以上のようにこれらの手法はいずれも制約面上を探索する構造を有しておらず，中には探索途中の解が許容解であることを保証されていないものもある。なお，制約面に乗る構造を有する手法として一般縮小勾配法があるが，不等号制約条件で制約面に抵触した場合の処理が十分に解決されているとはいえない。

活性でかつ有利な制約面上を探索する方法では制約面上に乗るために，その制約面についての方程式すなわち等式制約を解くことになるので，設計変数は自由に値の選べる探索変数と等式制約より定まる制約変数とに分かれる。よって，不等号制約条件と設計変数の中から，活性でかつ有利な制約面である等式制約とその制約変数を選択する必要が生じ，この選択をどのようにして行うかがこの方法の一番の問題点になる。

双対法では双対変数による探索で活性でかつ有利な制約面を最適解の段階で選んでいるが，この仕組みを利用すれば等式制約とその制約変数を選択するアルゴリズムが得られるのではないかと著者らは考え⁸⁾，次に述べる結果を得ることができた。

2. 制約式を選択

(1) 活性でかつ有利な制約面

最適設計問題は次の不等式制約付き最小化問題として表わされているとする。x は n 個の変数，g は r 個の不等号制約条件である。

$$\begin{cases} \min f(x) \\ \text{subj. to } g(x) \leq 0 \end{cases} \dots\dots\dots (1)$$

r 個の不等号制約条件の中から m 個 (m ≤ n) の制約条件を g_m(x) = 0 とする等式制約として選ぶと，n 個の変数 x は m 個の制約変数 x_m と n - m 個の探索変数 x_s とに分かれる。選択した等式制約 g_m は活性でかつ有利な制約面になっていなければならないが，g_m についての Lagrange 乗数を λ_m とすると，g_m，x_m，λ_m が具備すべき条件は式 (2) になる⁶⁾。

$$\begin{cases} g(x) = 0 \text{ の解が存在} & : \partial g_m / \partial x_m \cdot \det. \neq 0 \\ \text{他の制約式に抵触しない} & : g_{r-m}(x) \leq 0 \\ \text{有利な制約面} & : \lambda_m \geq 0 \end{cases} \dots\dots\dots (2)$$

式 (2) 第 1 行は g_m(x) = 0 の解が得られる，すなわち λ_m が存在することを強調したものであり，以降の記述

で λ_m の存在を前提としている場合には省略してある。

g_m 以外の不等号で満足している制約式 g_{r-m} についての Lagrange 乗数を λ_{r-m} とすると，式 (2) と合わせて g と λ について次の関係が成立する。

$$\begin{cases} g_m(x) = 0 \text{ については } \lambda_m \geq 0 \\ g_{r-m}(x) \leq 0 \text{ については } \lambda_{r-m} = 0 \end{cases} \dots\dots\dots (3)$$

λ_m は Lagrange 関数式 (4) の制約変数についての微分式 (5) より求める。

$$L(x, \lambda) = f(x) + \lambda^T g(x) \dots\dots\dots (4)$$

$$\begin{cases} \frac{\partial L(x, \lambda)}{\partial x_m} = \frac{\partial f(x)}{\partial x_m} + \lambda^T \frac{\partial g(x)}{\partial x_m} \\ = \frac{\partial f(x)}{\partial x_m} + \lambda_m^T \frac{\partial g_m(x)}{\partial x_m} = 0 \end{cases} \dots\dots\dots (5)$$

なお，式 (3) の関係を用いると，式 (4) より L(x, λ) = f(x) になり，等式制約では Lagrange 関数と目的関数は等しい。

選択した等式制約では式 (3)，(5) が成立しているが，これと不等号制約条件での最適解の必要条件である Kuhn-Tucker 必要条件式 (6) を比べると，等式制約では Lagrange 関数の探索変数についての微分式 (7) のみが成立していない。

$$\begin{cases} \frac{\partial L(x, \lambda)}{\partial x} = \frac{\partial f(x)}{\partial x} + \lambda^T \frac{\partial g(x)}{\partial x} = 0 \\ \frac{\partial L(x, \lambda)}{\partial \lambda} = g(x) \leq 0 \\ \frac{\partial L(x, \lambda)}{\partial \lambda} = \lambda = \lambda^T g(x) = 0, \lambda \geq 0 \\ \frac{\partial L(x, \lambda)}{\partial x_s} = \frac{\partial f(x)}{\partial x_s} + \lambda^T \frac{\partial g(x)}{\partial x_s} = 0 \end{cases} \dots\dots\dots (6)$$

$$\frac{\partial L(x, \lambda)}{\partial x_s} = \frac{\partial f(x)}{\partial x_s} + \lambda^T \frac{\partial g(x)}{\partial x_s} = 0 \dots\dots\dots (7)$$

等式制約すなわち活性でかつ有利な制約面上を式 (7) が成立するように x_s で探索していくと，探索が終了した段階では Kuhn-Tucker 必要条件が成立していることは明らかである。

(2) g ≤ 0 および λ ≥ 0 の満足

活性でかつ有利な制約面上を探索していくとき，他の制約式 g_{r-m} に抵触する (g_i > 0)，あるいは等式制約の中に不利な制約式が生ずる (λ_i < 0) 場合がある。また，探索で初期値 x を与えたときも同様のことが起こる可能性があるが，この g ≤ 0 および λ ≥ 0 が満足されない状態をどのようにして解決するかが大きな問題点である⁹⁾。

g ≤ 0 および λ ≥ 0 が満足されない場合に g_i > 0 と λ_i < 0 の解消は (8) のように処理するが，この処理による新たな等式制約 g_m は活性でかつ有利な制約面の条件式 (2) を満足する必要がある。

$$\begin{cases} g_i > 0 \text{ の制約式を } g_m \text{ に含める} \\ \lambda_i < 0 \text{ の制約式を } g_m \text{ からはずす} \end{cases} \dots\dots\dots (8)$$

(8) の第 1 行の処理を Fig.1 で説明すると，g_i に抵

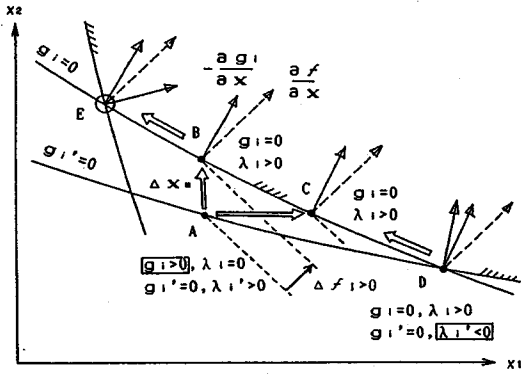


Fig. 1 Selection of active constraints.

触している A 点で g'_i を含む g_m に単純に g_i を追加し、この $g_m(x)=0$ を解くと D 点になり、D 点では $\lambda'_i < 0$ を生じる。これに対し g'_i を g_i で置き換えた $g_m(x)=0$ を解くとき、 x_2 が制約変数であれば B 点になり、 x_1 が制約変数であれば C 点になる。A 点から B 点または C 点に移行するとき、 $g_i > 0$ かつ $\lambda_i = 0 \rightarrow g_i = 0$ かつ $\lambda_i > 0$ の変化を生じ、 f の変化 $\Delta f > 0$ である。

A 点から B 点へ移行するときの線形近似した制約変数の変化 Δx_m は式 (9) であり、B 点の λ_m 推定値は式 (5) の $\partial L / \partial x_m = 0$ より定まることから式 (10) になる。

$$\Delta x_m = - \left(\frac{\partial g_m}{\partial x_m} \right)^{-1} g_m \dots\dots\dots (9)$$

$$\lambda_m = - \frac{\partial f}{\partial x_m} \left(\frac{\partial g_m}{\partial x_m} \right)^{-1} \dots\dots\dots (10)$$

式 (9), (10) を用いて Δf を表わすと式 (11) になり、(8) の第 1 行の処理による f の変化は $\Delta f = \lambda_m^T g_m$ になることがわかる。

$$\Delta f = \frac{\partial f}{\partial x_m} \Delta x_m = \frac{\partial f}{\partial x_m} \left\{ - \left(\frac{\partial g_m}{\partial x_m} \right)^{-1} g_m \right\} \dots\dots\dots (11)$$

$$= \lambda_m^T g_m$$

これらはいずれも A 点での値を用いて計算する。Fig. 1 では C 点よりも B 点の方が f が小さいので、 Δf は C 点よりも B 点の方が小さく、 $\Delta f_i = \lambda_i \cdot g_i$ において g_i は同じであるから、C 点より B 点のほうが λ_i が小さい。

$g_i > 0$ の制約式を g_m に含めていくことにより $\Delta f_i = \lambda_i \cdot g_i > 0$ の増加があるが、Fig. 1 の B 点のような活性化制約面を選ぶと $g_i > 0$ が存在しなくなり、これ以上 $\Delta f = \lambda_m^T g_m$ は大きくならず、 $\max \lambda_m^T g_m$ が達成されたと考えられる。すなわち抵触制約式が複数個あっても、 $g_i > 0$ が存在しなくなるまで、 $g_i > 0$ の制約式を g_m に含めていけばよい。

λ_m は式 (10) すなわち式 (5) より定まるが、 $\lambda_m \geq 0$ でなければならない。 $\lambda_{r-m} = 0$ を合わせると λ_m は λ で置き換えられるので、(8) の第 1 行 $g_i > 0$ の解消は式

(12) で表わされる。

$$\begin{aligned} & \max \lambda^T g \\ & \text{subj. to } \frac{\partial L}{\partial x_m} = \frac{\partial f}{\partial x_m} + \lambda^T \frac{\partial g}{\partial x_m} = 0 \dots\dots\dots (12) \\ & \lambda \geq 0 \end{aligned}$$

$g_i > 0$ の解消した B 点から x_1, x_2 の次元での最小点 E 点へは探索により活性化制約面 $g_i = 0$ 上を移行する。

(8) の第 2 行 $\lambda_i < 0$ の解消は式 (12) で $\lambda \geq 0$ を求めていることから達成される。すなわち $\lambda_i < 0$ になる制約式は g_m には含まれないことになる。仮に Fig. 1 の D 点で $\lambda'_i < 0$ が生じたとすると、式 (12) で $\lambda \geq 0$ を求めればこの g'_i は今までの g_m からはずされる。これにより活性化制約面 $g_i = 0$ 上を探索できるようになり、D 点から E 点へ移行する。

式 (12) で $\lambda \geq 0$ が得られれば式 (2) を満足する等式制約 g_m が得られたことになるので、式 (12) は $g \leq 0$ および $\lambda \geq 0$ が満足されていない場合の解決策になる。しかしながら、式 (12) だけでは変数 x のなかでどの x_j を制約変数 x_m にするかの区別がなされていないし、新たな g_m, x_m で $g_m(x)=0$ が解けるかどうかも確かめられていない。このためには次の問題点を解決しなければならない。

移行する点を求めるすなわち $g_m(x)=0$ を解くには制約式の集合 g_m に制約変数の集合 x_m を対応させる必要がある。よって $g_i > 0$ を g_m に含めるときの変数を制約変数 x_{m_j} として g_i に対応させるかを考えねばならない。 g_i を g_m に含める以前に x_{m_j} が探索変数であったときは g_i の単純な追加になり、 x_{m_j} に対応する制約式 g'_i がすでに存在していたときは g_i と g'_i の置き換えになる。これは x_{m_j} の選び方により移行する点が異なってくることを意味している。 x_{m_j} の選び方によっては $\lambda_i < 0$ を生じることがあり、場合によっては移行する点が求められないこともある。このようなことが起きない変数が x_{m_j} の候補であるが、候補となる x_{m_j} が複数個あるときはどれを選ぶかも決めねばならない。

このため、いかにして g_i に対応した x_{m_j} を選ぶかが重要な問題になってくる。選び方としては $\lambda_i < 0$ を生じないようにして、候補となる $\lambda_i \geq 0$ の中から λ_i 最小の x_{m_j} を選ぶことが考えられる。 λ_i 最小を選ぶ理由は λ_i が小さい方が f の増加が小さいので、その後の探索に都合がよいためである。 λ_m が求まることは式 (10) より $\partial g_m / \partial x_m$ の逆行列が存在し、式 (9) より Δx_m が得られることから $g_m(x)=0$ が解けることを意味している。よって $\lambda_i \geq 0$ が求められるような方法が必要である。

この点の解決のため式 (12) に着目してみると、式 (12) は λ について線形でありかつ線形計画問題に類似した

形になっていることから、式 (12) を線形計画問題として解くことが考えられる。しかしながら、式 (12) は $g \leq 0, \lambda \geq 0$ を満足する g_m, x_m の選択を目的としたものであり、式 (1) の不等式制約付き最小化問題の解を求めるものではない。

一方、 $\lambda \geq 0$ の条件で λ について maximum を求める方法に双対法があり、式 (12) は双対法とも関連した形である。よって双対法からアプローチして、 g_m, x_m の選択を線形計画問題として解くことを中心にした非線形最適化手法のアルゴリズムを構成してみることにする。

(3) 双対法と線形計画問題

双対法では双対変数 λ (Lagrange 乗数 λ) による探索で $g \leq 0$ および $\lambda \geq 0$ を満足する制約式を最適解の段階で選択しているが、この仕組みを利用して等式制約 g_m 、制約変数 x_m を選択する方法を考えてみる。

双対法は式 (13) で表わされる¹⁾。

$$\left. \begin{aligned} \max_{\lambda} h(\lambda) \\ \text{subj. to } \lambda \geq 0 \end{aligned} \right\} \dots\dots\dots(13)$$

ただし双対関数 $h(\lambda)$ は式 (14) であり、Lagrange 関数 $L(x, \lambda)$ が微分可能なら x について $\min L(x, \lambda)$ は式 (15) と等価である。

$$\left. \begin{aligned} h(\lambda) = \min_x L(x, \lambda) \\ = \min_x \{f(x) + \lambda^T g(x)\} \end{aligned} \right\} \dots\dots\dots(14)$$

$$\frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} + \lambda^T \frac{\partial g}{\partial x} = 0 \dots\dots\dots(15)$$

これより双対法は式 (16) で表わされる。

$$\left. \begin{aligned} \max_{\lambda} \{f(x) + \lambda^T g(x)\} \\ \text{subj. to } \frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} + \lambda^T \frac{\partial g}{\partial x} = 0 \\ \lambda \geq 0 \end{aligned} \right\} \dots\dots\dots(16)$$

式 (16) は λ について線形である。 x を固定すると、式 (16) 第 2 行がすべての x_j について必ず成立するとは限らないので、第 2 行の左半分のみを取り上げた式で置き換え、これを式 (17) とする。

$$\left. \begin{aligned} \max_{\lambda} \{f(x) + \lambda^T g(x)\} \\ \text{subj. to } \frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} + \lambda^T \frac{\partial g}{\partial x} \\ \lambda \geq 0 \end{aligned} \right\} \dots\dots\dots(17)$$

式 (17) 第 2 行を制約変数 x_m 、探索変数 x_s に分けると、 $\partial L / \partial x = 0$ は (18) のように実現される。これより式 (16) 第 2 行を成立させることができる。

制約変数については $\partial L / \partial x_m = 0$ となるように λ を定める。
探索変数については式 (17) とは別のところで $\partial L / \partial x_s = 0$ となるように x_s を定める。

$$\dots\dots\dots(18)$$

式 (17) を書き換えると、式 (19) になる。

$$\left. \begin{aligned} \max_{\lambda} \lambda^T g + f \\ \text{subj. to } -\lambda^T \frac{\partial g}{\partial x} + \frac{\partial L}{\partial x} = \frac{\partial f}{\partial x} \\ \lambda \geq 0 \end{aligned} \right\} \dots\dots\dots(19)$$

式 (19) は $f, g, \partial f / \partial x, \partial g / \partial x$ を係数とみなせば λ についての線形計画問題である。 $\partial f / \partial x$ は定数項であり、 $\partial L / \partial x$ は slack 変数に相当するが、 $\partial f / \partial x, \partial L / \partial x$ には非負条件はない。 $\Delta f = \lambda^T g$ であるので、式 (19) 第 1 行は $\Delta f + f$ になる。

式 (12) と式 (19) を比較すると、 x を固定すれば第 1 行は両者で同じことを意味しており、式 (19) 第 2 行で制約変数についてのみ $\partial L / \partial x_m = 0$ が成立するとしたものと式 (12) 第 2 行は同じである。式 (19) を解くことにより $\partial L / \partial x_m = 0$ となる $\lambda_m \geq 0$ (他の $\lambda_{r-m} = 0$) を定めることができれば、この λ_m は式 (12) の解になっている。これより $g_i > 0$ 解消のため式 (12) を解くことは双対法を線形計画問題として解く式 (19) で置き換えられる。

式 (19) を解くことにより $\lambda \geq 0$ となる g_m, x_m が選択できるので、 $\lambda_i < 0$ の解消は式 (19) で g_m, x_m を選択すれば自動的に解決される。また制約変数と探索変数の区別、 $g_m(x) = 0$ が解けるかどうかの問題点は式 (19) を解くための掃き出し計算で解決される。

(4) 掃出し計算

式 (19) を λ について解くには simplex 法に準じた掃出し計算を用いる。simplex tableau を Fig. 2 に示すが、slack 変数列には掃き出し開始時の基底変数が入れている。No. 列は基底変数になった λ_i または $\partial L / \partial x_{s_i}$ の番号である。掃出し列および掃出し行の選択は $g_i > 0$ の解消に対応して (20) のようになる。

$-g_i < 0$ ($g \leq 0$ に抵触) の λ_i を $\lambda \geq 0$ となるように掃出すが、 x についての $\min L(x, \lambda)$ を考慮する。
ただし $\partial f / \partial x, \partial L / \partial x$ には非負条件なし。

$$\dots\dots\dots(20)$$

式 (20) を具体化したのが (21), (22) である。掃出し列 k 列の選択は g_i の値を基準化しておいて、式 (21) で選択すればよい。 $g_i > 0$ 解消のためには任意の順序で $g_i > 0$ を掃出し列とすればよいが、抵触の最も大きい $g_i > 0$ を k 列として選ぶ方が $g_i > 0$ 解消の効率はよい。掃出し行 p 行の選択は定数項 $\partial f / \partial x$ 、slack 変数 $\partial L / \partial x$ に非負条件を設けていないので、 λ_k は式 (22) になる。 $\lambda_w \geq 0$ の中から最小の j 行を λ_k として選んでいるのは、 $\Delta f_k = \lambda_k \cdot g_k$ より同じ g_k なら λ_k の小さいものを選んだ方が f の増加が小さくなり、これは双対関

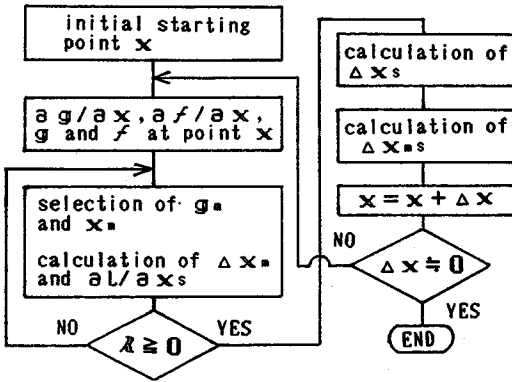


Fig. 5 A general flow chart.

3. 探索プログラム

(1) プログラム上の要点

上記2. で制約式選択の方法について示したが、Fig. 4の概要フローチャートをもとに探索プログラムを作成することができる。基本的な仕組みは概要フローチャートで十分に示されているが、探索プログラム化していくときに次の点を考慮した。

制約面上を探索しているとき、探索変数の個数をなるべく少なくした方が一般に効率がよいので、等式制約に含まれる制約式の個数はなるべく多い方がよい。このため掃出し計算での g_m, x_m の選択のとき、等式制約に含まれている制約式を最初に掃出しておいてから、次に抵触した制約式を掃出すようにしている。これは等式制約に含まれる制約式を一定に保つ効果もある。なお、探索開始時の設計変数 x の初期値が許容領域内にあるかどうかは問題にしていないが、多くの制約式に抵触している初期値の方が最初の等式制約に含まれる制約式の個数が多くなり、効率がよくなる。これは許容領域内からは探索により、許容領域外からは制約式の選択により、活性な制約面に到達する仕組みを用いているためである。

式 (25) より得られた Δx_s について $x_s = x_s + \Delta x_s$ として探索すると、新たな制約面に抵触することがあるが、 Δx_s の絶対値が大きいと抵触制約面の下に奥深く入ることになる。この抵触を解消しようとするとき、今まで乗っていた制約面から離れて抵触制約面上の点を選択する可能性が大となり、今まで乗っていた制約面と抵触制約面の交点を選択する可能性は小さくなる。もしこの交点を選択する必要がある場合に（たとえば交点が最適解のとき）これでは不都合を生じる。この交点を選択する可能性を高めるため、式 (28) のように得られた Δx_s に小さな step 幅 α を掛けて少しずつ進む方法を取っている。

$$x_s = x_s + \alpha \cdot \Delta x_s \dots \dots \dots (28)$$

step 幅 α として、制約式に抵触しているときは 0.1、活性な制約面を得れば 0.2 のように変化させた値を用いている。簡易な Δx_s が式 (25) より常に得られるので、 Δx_s に方向の悪さがあっても掃出し計算ごとに Δx_s を計算し直すことで補うことができる。

(2) 探索例

上記2. の考え方に基づいた探索プログラムを作成するため、床版断面を与件とした単純活荷重合成桁の主桁断面決定を具体例として取り上げた。設計変数は腹板高 h_w 、腹板厚 t_w 、圧縮フランジ幅 b_c 、圧縮フランジ厚 t_c 、引張りフランジ幅 b_t 、引張りフランジ厚 t_t の6個とし、鋼桁断面積を目的関数（断面変更点のある場合は鋼桁体積）としたときの不等式制約付き最小化問題は式 (29) である。

$$\min f(x) = h_w \cdot t_w + b_c \cdot t_c + b_t \cdot t_t$$

subj. to

- $g_{sca} = \sigma_{sc} + \sigma_{vc} - \sigma_{ca} \leq 0$
- $g_{ser} = \sigma_c - \sigma_{er} \leq 0$
- $g_{bco} = b_{co} - b_c \leq 0$
- $g_{iv} = \mu \cdot M_o - I_v \leq 0$
- $g_{svc} = -\sigma_{vc} \leq 0$
- $g_{sta} = \sigma_{st} + \sigma_{vt} - \sigma_{ta} \leq 0$
- $g_{bto} = b_{to} - b_t \leq 0$
- $g_{tww} = t_{wo} - t_w \leq 0$
- $g_{tww} = h_w / r_w - t_w \leq 0$
- $g_{tco} = t_{co} - t_c \leq 0$
- $g_{tcr} = b_c / r_c - t_c \leq 0$
- $g_{tto} = t_{to} - t_t \leq 0$
- $g_{ttr} = b_t / r_t - t_t \leq 0$
- $g_{twn} = t_w - t_{wn} \leq 0$
- $g_{tcn} = t_c - t_{cn} \leq 0$
- $g_{tin} = t_t - t_{tn} \leq 0$

(29)

ただし式 (29) の16個の制約条件は

- g_{sca} : 圧縮縁許容応力度
- g_{ser} : 架設時許容応力度
- g_{bco} : 圧縮フランジ最少幅
- g_{iv} : たわみ制限
- g_{svc} : 中立軸が鋼桁内
- g_{sta} : 引張り縁許容応力度
- g_{bto} : 引張りフランジ最少幅
- g_{tww} : 腹板最少板厚
- g_{tww} : 腹板最大幅厚比
- g_{tco} : 圧縮フランジ最少板厚
- g_{tcr} : 圧縮フランジ最大幅厚比
- g_{tto} : 引張りフランジ最少板厚
- g_{ttr} : 引張りフランジ最大幅厚比

- g_{tw1} ：腹板最大板厚
- g_{tc1} ：圧縮フランジ最大板厚
- g_{tt1} ：引張りフランジ最大板厚

作成した探索プログラムを用い、この鋼桁断面の最適解を設計曲げモーメント $M = 50 \sim 1200 \text{ MN} \cdot \text{m}$ ($50 \text{ MN} \cdot \text{m}$ おき) について求めた。活荷重合成桁では設計曲げモーメントは合成前荷重に対する M_s と合成後荷重に対する M_o から成り立っているが、図示するときの便利さを考慮して、 M を与えて M_s と M_o に分割するようになっている。分割するには $M = M_s + M_o$ と M_o/M_s の関係を表わす近似式を作成し、これを用いる。

最適解（最適桁高）がどのような制約面上にあるかを図示するため、活性な制約面の状況を探索プログラムを若干修正して調べた。活性な制約面の状況を調べるにはある変数 x_j を指定し、この x_j を一定の間隔で変化させて活性な制約面上での g, λ, x 等を計算する。このとき他の変数のうちのどれかが一定値を保つように指定することも可能である。 x_j を指定するには x_j が制約変数でも探索変数でもないようにし、 x_j に値を与えればよい。 x_j を指定したときに得られる活性な制約面は x_j を指定したときの最適解になっている。

制約面が変化したとき、境界をはさんで指定した x_j で step 幅を小さくしながら、前進・後退を繰り返すことにより境界が求められる。修正したプログラムには一定間隔で g, λ, x 等を計算し、かつ制約面が変化すると制約面の境界を求める機能がある。このプログラムを用いると、Fig. 6 の活性な制約面を簡単に求めることができる。

Fig. 6 は曲げモーメント M に対し h_w を変化させて、制約面の状況を調べた図に最適桁高を併記したものである。この例では6個の変数のうち1個が探索変数になるのが通常のケースなので、等式制約には5個の制約式が含まれる場合が多い。Fig. 6 では図が複雑になるのを避けるため、主としてフランジ幅を規定する制約式2個と腹板厚の制約式1個のみが表示されており、主としてフランジ厚を規定する制約式2個は省略されている。

Fig. 6 に記入されている制約式名は活性な（かつ有利な）制約面であり、設計曲げモーメントと桁高に対し、どのような制約条件で桁断面の設計を行ったらよいかが表示されている。この図にフランジ断面積の等高線を併記すると、フランジ断面積決定の目安にすることができ、さらに発展させるとノモグラムの使用が可能になる。

最適桁高は同一の制約面上なら滑らかな曲線であり、制約面が異なると折線になるが、この様子が Fig. 6 に示されている。この図に目的関数の等高線を併記すると、実用上は最適解とみなし得る範囲が明らかになる。最適解の近辺では活性な制約面上の目的関数の勾配は一般に

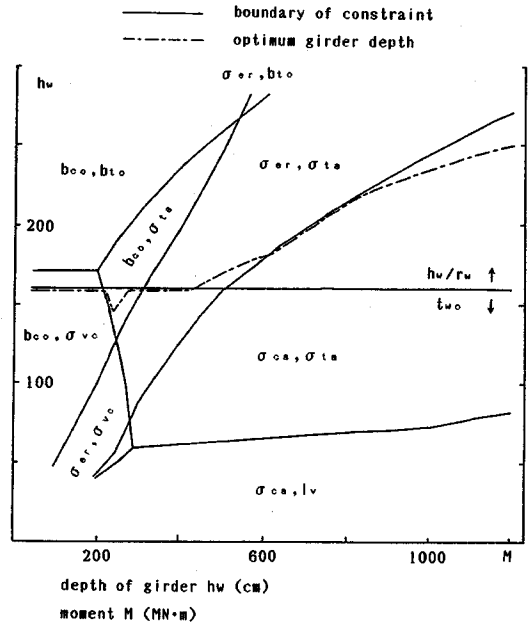


Fig. 6 Active constraint surfaces.

緩いことから、最適解近辺での設計変数には選択の幅があるのが普通である。プレートガーダーの場合、 g_{suc} (中立軸が鋼桁内) の制約条件が存在しないため、設計曲げモーメントが小さくなるにつれ最適桁高も小さくなる。

ここに挙げた探索例は比較的簡単な例であるが、設計変数と制約条件の個数が多い場合にも本法では活性な制約面を確実に選択することができる。通常最適解の探索だけでなく、制約面の状況を調べる等の設計的な利用が容易に行えることも本法の特色の1つである。

(3) 収束状況

活性な制約面上に乗せているため本法による解の精度は優れており、目的関数について計算機の有効桁数（単精度6桁）の末尾が変化しなくなる状態を最適解としたが、実用上はこのような精度は必要なく、最適解との差が0.01%未満になり、かつ抵触した制約式がない状態を収束とみなすことにした。効率性を調べるため、上記(2)の探索例で断面変更点の個数 $N = 0 \sim 4$ 、設計曲げモーメント $M = 100 \sim 1000 \text{ MN} \cdot \text{m}$ ($100 \text{ MN} \cdot \text{m}$ おき) について、収束に要する $\partial g / \partial x$ の計算回数を Table 1 にまとめた。

N に対応した設計変数の個数 n と制約式の個数 r が増加すると、抵触が多くなること、探索方向ベクトルに難しさが生じてくること等により、 $\partial g / \partial x$ 回数が増加してくるが、大幅な増加ではない。本法では一次元探索を用いていないので、 g 評価回数は $\partial g / \partial x$ 回数のみを考慮すればよい。

Table 1 Iteration number of $\partial g/\partial x$.

N	n	r	range	mean
0	6	16	4~9	7.4
1	11	30	4~16	11.6
2	16	44	8~19	13.3
3	21	58	8~25	14.9
4	26	72	9~30	16.5

N : number of alteration points of the girder section

n : number of design variables

r : number of constraints

$\partial g/\partial x$ 回数を調べたときの本法での代表例として、 $N=0: M=700 \text{ MN}\cdot\text{m}$ 、 $N=4: M=600 \text{ MN}\cdot\text{m}$ を取り上げ、収束状況を目的関数 f 、腹板高 h_w で Fig. 7 に示す。 f の初期値は小さめになっており、2~3回の反復で活性化制約面の近くに到達し、その後は一様な収束になる。この収束状況は本法の安定性を示しているといえる。

最小化法として可変計量法を用いた罰金法（外点法）について、一次元探索での g 評価回数も含めて換算した $\partial g/\partial x$ 回数を調べると、たとえば $N=0$ では1応答面の探索だけで7~9回を要しており、本法ではこの回数だけで最適解との差0.01%未満に収束することができる。両者とも初期値は許容領域内である必要はないが、罰金法での応答係数の調整のような技巧的な処理を本法は必要としていない。罰金法では活性化制約面上に乗せることが難しいことから、探索例では最適解との差0.1%程度の収束であり、解の精度は本法の方が優れている。

この理由は、罰金法では活性化制約面上に乗せる操作と最小点の探索を探索方向ベクトルのみに依存しているのに対し、本法では活性化制約面を掃出し計算で求め、得られた活性化制約面上での最小点を探索で求める、というように制約条件の満足と探索の2つに機能を分けて

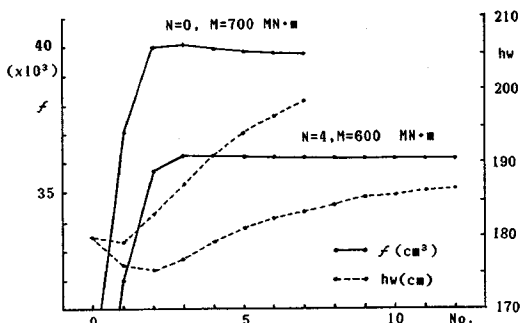


Fig. 7 Iteration number of $\partial g/\partial x$.

いるためである。上記の最適解との差は目的関数についてであり、最適解の近辺では目的関数の差に比べ、設計変数の差はかなり大きくなる。

(4) 最適化手法としての特色

ここで示した方法は設計問題での要求に対応できる手法として開発されたが、最適化手法としてみると次の特色がある。

- a) アルゴリズムは単純明解であり、特別な技巧は必要としない。
- b) 活性化制約面に沿って行われる探索が掃出し計算を中心とした tableau の操作のみで達成される。
- c) $\partial L/\partial x_s = 0$ となるような探索により探索終了判定で Kuhn-Tucker 必要条件を満足していることが確認できる。
- d) 初期値が許容領域内にあるかどうかは問題にしない。
- e) 等式制約を扱っているのに、不等号条件だけでなく等号条件を含む場合にもなら支障はない。
- f) 最適解が許容領域の頂点にないときでも、逐次線形計画法のような解の振動を生じない。
- g) 双対変数ではなく設計変数で探索しているのに、双対法と比べ探索径路が把握しやすい。
- h) 縮小勾配を用いて活性化制約面上を探索しようとしている点で共通性をもつ GRG と比較する¹¹⁾。GRG では不等号制約式に slack 変数を用いて等号化し、slack 変数が0となる制約式を活性化制約式とし、変数の境界からの距離で変数を順序付けして、この順序で基底変数、非基底変数を区別している。本法では tableau の掃出し列、掃出し行の選択で活性化制約式と基底変数を選んでいく。したがって、GRG では活性化制約式と基底変数の選択が変数の順序付けに集約されており、この点のアルゴリズムに明確さが欠け、また、slack 変数で等号化しているため、変数、 $\partial L/\partial x$ の次元が大きくなっており、これらの点で本法の方が優れていると考えられる。

4. まとめ

(1) 双対法を線形化して simplex 法に準じた掃出し計算を行うことにより、活性でかつ有利な制約面となるような等式制約とその制約変数を、不等号制約条件と設計変数の中からどのようにして選択するかの問題点を解決することができた。

(2) 掃出し計算のとき $g_m(x) = 0$ を Newton 法で解くときの Δx_m が得られることから、掃出し計算の反復だけで活性でかつ有利な制約面を得ることができる。

(3) 上記の方法に基づく探索プログラムを作成し、活荷重合成桁の主桁断面決定を応用例として探索した結

果から、著者らが開発した方法による探索プログラムが十分実用性のあることを確認した。

(4) 活性でかつ有利な制約面を探索する方法には多くの長所がある。制約式と変数の選択に関する問題点が解決できたことから、著者らが開発した方法は今後有効に使えるものと思われる。

参 考 文 献

- 1) 志水清孝：システム最適化理論，コロナ社，pp.5~101, 1976.
- 2) Gallagher, R. H. and Zienkiewicz, O. C. ed. : Optimum Structural Design, John Wiley & Sons, Inc., pp.109~177, 1973.
- 3) L. C. W. ディクソン (松原正一訳)：非線形最適化計算法，培風館，pp.38~71, 1974.
- 4) O. L. マンガサリアン (関根智明訳)：非線形計画法，培風館，pp.90~128, 1972.
- 5) U. Kirsch (山田善一・大久保禎二監訳)：最適構造設計，丸善，pp.132~176, 1983.
- 6) 平田恭久・伊藤文人：等式制約探索法について，土木学会第37回年次学術講演会講演概要集，第1部，1-20.
- 7) 平田恭久・伊藤文人：等式制約探索法による探索プログラム，土木学会第38回年次学術講演会講演概要集，第1部，1-226.
- 8) 平田恭久・伊藤文人：等式制約探索法と双対探索法の対応について，土木学会第39回年次学術講演会講演概要集，第1部，1-208.
- 9) 平田恭久・伊藤文人：等式制約法での制約式の取捨選択について，土木学会第40回年次学術講演会講演概要集，第1部，1-162.
- 10) 平田恭久・伊藤文人：シンプレックス・タブローを用いた最適解の探索，土木学会関東支部第14回年次研究発表会，第1部，1-3.
- 11) Reklaitis, G. V., Ravindran, A. and Ragsdell, K. M. : Engineering Optimization, Wiley-Interscience, pp.381~402, 1983.

(1986.11.7・受付)