

## ベクトルペナルティ型拡張ラグランジュ関数の 逐次二次計画法への適用†

田村正義<sup>††</sup> 野中久典<sup>†††</sup> 小林康弘<sup>†††</sup>

非線形最適化手法としてシトコウスキイの提案した逐次二次計画法は、高い評価を受け、ソフトウェアを実現する上で、しばしば採用されている。その大きな理由は、メリット関数（直線探索の目的関数）として用いられている拡張ラグランジュ関数が、Maratos effect を防止できるからである。Maratos effect とは、探索点のわずかな改良を繰り返す、効率が悪くなる現象を示す。拡張ラグランジュ関数では、制約条件の重みであるペナルティパラメータはスカラである。本論文では、まず上記の拡張ラグランジュ関数が制約条件の勾配が互いに大きく異なる場合などに効率が悪いことがあることを指摘する。次に、ペナルティパラメータの値を各制約条件に対応して決められるようにベクトルにしたベクトルペナルティ型拡張ラグランジュ関数を提案する。さらに、ベクトルペナルティ型拡張ラグランジュ関数をシトコウスキイの提案した逐次二次計画法に適用するためのベクトルペナルティの更新方法と、それを用いたときの収束の保証について論じる。そのために、根拠となる定理を証明する。また、その有効性を確認するためにコンピュータプログラムを作成し、非線形最適化問題をを用いて行った数値実験結果についても述べる。

### 1. はじめに

近年、計算機利用技術に対するニーズの拡大に伴い、解くべき非線形最適化問題の規模も大きく、複雑になってきている。従来からある手法および、プログラムにとって、そのような問題は、場合によっては解くのが難しくなるので、それらの陳腐化が指摘されるようになる。そのため、より有効な非線形最適化手法とその手法を用いたコンピュータプログラムの開発が望まれている。例えば、構造解析の分野では、規模の大きい解析の対象物を自動設計するために、高度な最適化技術が求められている<sup>9)~11)</sup>。このように、実用的な問題において、非線形最適化に対する要求は高まっている。

非線形最適化手法としてシトコウスキイの提案した手法<sup>9)</sup>が高い評価を受けている。これは逐次二次計画法である。特に、その中で、メリット関数（直線探索時に目的関数の小ささと制約条件の満足度を同時に評価するための関数）として用いられている拡張ラグランジュ関数は、逐次二次計画法に限らず乗数法などにも広く用いられるほど評価が高い。その拡張ラグランジュ関数では、制約条件の重みであるペナルティパラ

メータはスカラである。これをスカラペナルティ型拡張ラグランジュ関数と呼ぶ。

本論文では、拡張ラグランジュ関数のペナルティパラメータをベクトルにすることを提案する。その拡張ラグランジュ関数の一改良案をベクトルペナルティ型拡張ラグランジュ関数と呼ぶ。しかし、単にメリット関数を交換するだけでは、ベクトルペナルティ型拡張ラグランジュ関数をシトコウスキイの提案した逐次二次計画法に適用することはできない。ベクトルペナルティ型拡張ラグランジュ関数をメリット関数として用いた逐次二次計画法の収束を保証する必要がある。そのためには、収束を保証するための定理を証明するだけでなく、その定理の仮定を満足するベクトルペナルティの更新方法を構成することが必要である。

本論文の提案は、逐次二次計画法という枠組の中でスケールリングをある程度自動的に行うことを目的としている。これにより、問題を定義する者の負担を軽くすると共に、逐次二次計画法の適用範囲を拡大させる。

2章で、基礎知識として、解くべき非線形最適化問題の定義と逐次二次計画法の概略を述べる。3章でスカラペナルティ型拡張ラグランジュ関数、スカラペナルティの更新方法とその欠点を述べる。4章で、ベクトルペナルティ型拡張ラグランジュ関数とベクトルペナルティの更新方法を提案し、これを用いたときの逐次二次計画法の収束を保証する定理を証明する。5章で、数値実験結果について言及する。

† Augmented Lagrangean Function with Penalty Vector for Sequential Quadratic Programming by MASAYOSHI TAMURA (Artificial Intelligence Department, Software Development Center, Hitachi Ltd.), HISANORI NONAKA and YASUHIRO KOBAYASHI (1st Department, Energy Research Laboratory, Hitachi Ltd.).

†† (株)日立製作所ソフトウェア開発本部 AI 設計部

††† (株)日立製作所エネルギー研究所第1部

2. 基礎知識

2.1 非線形最適化問題

本論文では、非線形最適化問題を次のように記述する。

$$\text{最小化 } f(x), \tag{2.1}$$

$$\text{制約条件 } g_i(x)=0 \quad (i \in ME), \tag{2.2}$$

$$g_i(x) \geq 0 \quad (i \in MI). \tag{2.3}$$

ここで、 $x \in R^n$  であり、 $f(x)$ ,  $g_i(x)$  の中の一つ以上が非線形であることを許すが、二階微分可能であることを仮定する。さらに、 $f(x)$  は凸関数、許容領域（すべての制約条件を満足する領域）は凸領域であることも仮定する。 $f(x)$  を目的関数と呼び、 $g_i(x)$  を制約関数と呼ぶ。また、 $MI \cup ME$  の要素数を  $m$  とする。

この非線形最適化問題に対するラグランジュ関数は次のとおりである。

$$L(x, v) = f(x) - \sum_{i \in MI \cup ME} v_i g_i(x). \tag{2.4}$$

ここで、 $v_i$  は非線形最適化問題の制約条件  $i$  に対応するラグランジュ乗数である。

2.2 逐次二次計画法

ここでは、シトコウスキイの提案した逐次二次計画法<sup>1)</sup>について述べる。

逐次二次計画法のアルゴリズムを図1に示す。

①では、②で求める探索方向ベクトル  $d$  の大きさがあらかじめ設定された微小正数  $\epsilon$  よりも小さくなるまで反復を繰り返す。このとき、探索点  $x$  と二次計画問題を解くことによって求めた二次計画問題の制約条件に対応するラグランジュ乗数  $u$  のペアが一次のキューン・タッカー条件を満足している。キューン・タッカー条件の中にラグランジュ関数  $L$  の  $x$  に関する勾配の大きさが0になるという条件である。

②では、非線形最適化問題の近似問題である二次計画問題を解くことによって、探索方向ベクトル  $d$  を決める。二次計画問題の目的関数は二次式、制約条件は

一次式である。

③では、メリット関数の中のペナルティパラメタを更新する。この更新によって、探索方向がメリット関数の降下方向になっていることが保証できる。

④では、探索方向にどれだけ進むかを直線探索によって決める。直線探索時、探索方向ベクトルに対して、どれだけの比率分進めたかを探索幅と呼ぶ。直線探索時、探索方向ベクトル分進めれば、探索幅は1であり、半分だけ進んだときには、探索幅は0.5である。

⑤の近似ヘッセ行列  $B$  の更新とは、(2.4)式に示したラグランジュ関数のヘッセ行列に対する  $B$  の近似精度を改良することを意味する。この更新には、パウエルの修正 BFGS 公式<sup>1)</sup>を用いている。②の条件を満足するまで①～⑤の処理を繰り返す。

各反復で解く二次計画問題は次のとおりである。

$$\text{最小化 } \frac{1}{2} d^T B d + \nabla f(x)^T d, \tag{2.5}$$

$$\text{制約条件 } g_i(x) + \nabla g_i(x)^T d = 0 \quad (i \in ME), \tag{2.6}$$

$$g_i(x) + \nabla g_i(x)^T d \geq 0 \quad (i \in MI). \tag{2.7}$$

ここで、 $\nabla$  は勾配、 $^T$  は転置、近似ヘッセ行列  $B$  および探索点  $x$  は各反復時の値を表す。

この二次計画問題に対応するラグランジュ関数は次のとおりである。

$$L(d, u) = \frac{1}{2} d^T B d + \nabla f(x)^T d - \sum_{i \in MI \cup ME} u_i (g_i(x) + \nabla g_i(x)^T d). \tag{2.8}$$

ここで、 $u_i$  は二次計画問題の制約条件  $i$  に対応するラグランジュ乗数である。

シトコウスキイの提案した逐次二次計画法では、二次計画法の解  $(d, u)$  に現在の探索点と近似ラグランジュ乗数の組  $(x, v)$  に対して、 $(d, u-v)$  を探索方向ベクトルとして直線探索を行う。つまり、直線探索時に探索点  $x$  と近似ラグランジュ乗数  $v$  を同時に変化させる。この点に関しては、山下<sup>3)</sup>の方法のように直線探索前にまず適切なラグランジュ乗数の値  $v^*$  を求めた後で  $x$  だけを変化させる方法もある。

メリット関数として、正確なペナルティ関数<sup>7)</sup>などもあるが、シトコウスキイは拡張ラグランジュ関数を採用している。

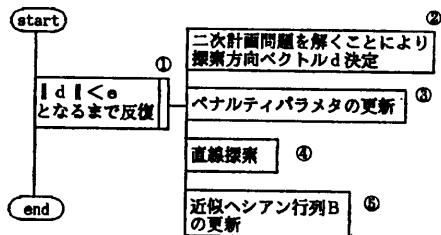


図1 逐次二次計画法のアルゴリズム  
Fig. 1 Algorithm of sequential quadratic programming method.

### 3. スカラペナルティ型拡張ラグランジュ関数

シトコウスキイが提案した拡張ラグランジュ関数を本論文では、スカラペナルティ型拡張ラグランジュ関数と呼ぶ。この関数の逐次二次計画法への適用について述べる。

#### 3.1 スカラペナルティ型拡張ラグランジュ関数

スカラペナルティ型拡張ラグランジュ関数は(3.1)式のとおりである。

$$\begin{aligned} \Phi(x, v; r) = & f(x) \\ & - \sum_{i \in M1} \left\{ v_i q_i(x) - \frac{1}{2} r q_i(x)^2 \right\} \\ & - \sum_{i \in M2} \frac{v_i^2}{2r}, \end{aligned} \quad (3.1)$$

ただし,  $M1 = \{i \in MI; q_i(x) \leq v_i/r\} \cup ME$ ,

$$M2 = \{i \in MI; q_i(x) > v_i/r\}.$$

目的関数と制約関数をスカラペナルティ  $r$  を用いて結合している。拡張ラグランジュ関数は制約条件の境界で一階導関数が連続になるように作られている。制約条件の境界とは,  $q_i(x) = 0$  となる点  $x$  の集合のことである。正確なペナルティ関数に比べて, 制約条件の境界でメリット関数の変化をなめらかにすることによって Maratos effect<sup>8)</sup> を防止している。

Maratos effect とは, 直線探索時, 探索幅が小さく, 探索点の改良を効率良く行えないことを意味する。探索幅とは, 直線探索で, 探索方向ベクトルの何割分進めたかを示す値である。曲面である制約条件の境界に沿って探索が進むときなどに, この現象が起きることがある。

#### 3.2 スカラペナルティの更新方法

スカラペナルティ型拡張ラグランジュ関数のスカラペナルティの更新方法は次のとおりである。

各反復で,  $\|u-v\|$  が十分小さいときには,  $r$  は更新しない。 $\|u-v\|$  が十分小さくないときには, 次のように  $r$  を更新する。ここで,  $u$  は, 二次計画問題の制約条件に対応するラグランジュ乗数であり,  $v$  は, 元々の非線形最適化問題の制約条件に対応する近似ラグランジュ乗数である。

次式により,  $\delta, \varepsilon$  を更新する。

$$\delta = \min(d^T B d / \|d\|^2, \delta), \quad (3.2)$$

$$\varepsilon = \|d\|^2 / \|u-v\|^2. \quad (3.3)$$

次の式を満足する最小の正数  $j$  を選ぶ。

$$\frac{1}{r^j} < \frac{1}{4} \varepsilon \delta \left(1 - \frac{\delta}{4}\right). \quad (3.4)$$

スカラペナルティ  $r$  を次式で更新する。

$$r = \max(r, \bar{r}^j). \quad (3.5)$$

ただし,  $\bar{r}$  および,  $r$  の初期値は 2,  $\delta$  の初期値は 1 である。

これにより, 二次計画問題を解くことによって求めた探索方向ベクトル  $(d, u-v)$  がスカラペナルティ型拡張ラグランジュ関数の降下方向になることを保証するのに十分な大きさに, スカラペナルティ  $r$  を更新している。このことから, 逐次二次計画法が最適点を発見することを保証できる。

#### 3.3 スカラペナルティ型拡張ラグランジュ関数の欠点

スカラペナルティ型拡張ラグランジュ関数を用いると, 全制約条件に共通のペナルティパラメタが用いられていることが最適化失敗の原因となり, 問題が解けなくなる例が見出せる<sup>9)</sup>。これらの問題は, 制約条件の勾配が互いに大きく異なるため, スカラペナルティでは対応しきれなくなっている。

この現象を回避するために, 従来, 非線形最適化問題を解く際, 目的関数や制約条件のバランスを考えて, スケーリングと呼ばれる処理を施さなければならなかった。スケーリングには, (1)目的関数や制約条件の各々を定数倍する, (2)変数変換する等の方法があった。これは, 言わば最適化という処理の外側でアドホックに行われる処理である。したがって, 例えば初期状態では良いスケーリングが, 反復が進むと望ましくないスケーリングになってしまうこともあった。

### 4. ベクトルペナルティ型拡張ラグランジュ関数の逐次二次計画法への適用

#### 4.1 ベクトルペナルティ型拡張ラグランジュ関数

3.3 節で述べたスカラペナルティ型拡張ラグランジュ関数が, 数値計算上潜在的に抱えている欠点を解決するための改良を提案する。改良とは, ペナルティパラメタ  $r$  をスカラからベクトルに変え, 各要素  $r_i$  が各制約条件に対応して値が変化できるようにすることである。この改良結果であるベクトルペナルティ型拡張ラグランジュ関数を(4.1)式に示す。

$$\Phi(x, v; r) = f(x)$$

$$- \sum_{i \in M1} \left\{ v_i q_i(x) - \frac{1}{2} r_i q_i(x)^2 \right\}$$

$$-\sum_{i \in M2} \frac{v_i^2}{2r_i} \tag{4.1}$$

ただし,  $M1 = \{i \in MI; g_i(x) \leq v_i/r_i\} \cup ME$ ,  
 $M2 = \{i \in MI; g_i(x) > v_i/r_i\}$ .

4.2 ベクトルペナルティ更新方法

ベクトルペナルティ型拡張ラグランジュ関数のベクトルペナルティ  $r$  の更新は次のようにする。

各反復で,  $u_i = v_i$  のときには  $r_i$  は更新せず,  $u_i \neq v_i$  のときに次に示すように  $r_i$  を更新する。ここで,  $u_i$  は, 二次計画問題の制約条件  $i$  に対応するラグランジュ乗数であり,  $v_i$  は, 元々の非線形最適化問題の制約条件  $i$  に対応する近似ラグランジュ乗数である。

次式により,  $\delta, \varepsilon_i$  を更新する。ここで,  $m$  は制約条件の個数である。

$$\delta = \min(d^T B d / \|d\|^2, \delta). \tag{4.2}$$

$$\varepsilon_i = \|d\|^2 / \{m(u_i - v_i)^2\}. \tag{4.3}$$

次の式を満足する最小の正数  $j$  を  $j_i$  とする。

$$\frac{1}{\bar{r}^j} < \frac{1}{4} \varepsilon_i \delta \left(1 - \frac{\delta}{4}\right). \tag{4.4}$$

ベクトルペナルティの各要素  $r_i$  を次式で更新する。

$$r_i = \max(r_i, \bar{r}^{j_i}). \tag{4.5}$$

ただし,  $\bar{r}$  および,  $r_i$  の初期値は 2,  $\delta$  の初期値は 1 である。

この更新は, 4.3 節に示す定理の仮定を満足するように構成してある。これにより, 二次計画法を解くことによって求めた探索方向ベクトル  $(d, u-v)$  が, ベクトルペナルティ型拡張ラグランジュ関数の降下方向になることを保証するのに十分な大きさにベクトルペナルティ  $r$  を更新している。このことから, 逐次二次計画法が最適点を発見することを保証できる。

4.3 収束を保証する定理

ベクトルペナルティ型拡張ラグランジュ関数をメリット関数として用いるために収束定理を証明する必要がある。この定理により, ベクトルペナルティ型拡張ラグランジュ関数をメリット関数として用いた逐次二次計画法の収束性を保証する。

〈記法〉

$$J = \{i | g_i \leq v_i/r_i, i \in MI\} \cup ME. \tag{4.6}$$

$$K = \{i | i \in J, i \in MI\}. \tag{4.7}$$

$$A = A(x) = (\nabla g_i(x)) \quad (i \in MI \cup ME). \tag{4.8}$$

$$\bar{v} = (\bar{v}_1, \dots, \bar{v}_m)^T, \tag{4.9}$$

ただし,  $\bar{v}_i(x) = v_i, \quad (i \in J)$

$\bar{v}_i(x) = 0. \quad (i \in K)$

$$g = g(x) = (g_1(x), \dots, g_m(x))^T. \tag{4.10}$$

$$\bar{g} = \bar{g}(x) = (\bar{g}_1(x), \dots, \bar{g}_m(x))^T, \tag{4.11}$$

ただし,  $\bar{g}_i(x) = g_i(x) \quad (i \in J)$ ,

$\bar{g}_i(x) = 0 \quad (i \in K)$ .

$$\bar{g}' = \bar{g}'(x) = (\bar{g}'_1(x), \dots, \bar{g}'_m(x))^T, \tag{4.12}$$

ただし,  $\bar{g}'_i(x) = g'_i(x), \quad (i \in J)$

$\bar{g}'_i(x) = v_i/r_i. \quad (i \in K)$

$$\bar{g}(x) = (r_1 \bar{g}_1(x), \dots, r_m \bar{g}_m(x))^T. \tag{4.13}$$

$$\nabla \Phi(x, v; r) = \begin{pmatrix} \nabla f(x) - A(x)(\bar{v} - \bar{g}(x)) \\ -\bar{g}'(x) \end{pmatrix}. \tag{4.14}$$

$R$ : 対角要素が  $r_i$  で, 他の要素がすべて 0 の行列.  $\tag{4.15}$

$E$ : 対角要素が  $\varepsilon_i$  で, 他の要素がすべて 0 の行列.  $\tag{4.16}$

$$t = \|B\| + \|A^T R A\|. \tag{4.17}$$

〈定理〉

(仮定 1)  $\varepsilon, \delta$  と  $r$  は, 正数で,  $\delta < 4$  であるとす

る。  
 (仮定 2) 正定値行列  $B$  に対して, (2.5)~(2.7) 式に示した二次計画問題は解けるものとする。つまり, 二次計画問題の解  $(d, u)$  は, 二次計画問題のキューン・タッカー条件を満足するものとする。

(仮定 3)  $g_i(x) + \nabla g_i(x)^T d = 0$  のとき, かつ, そのときのみ  $g_i(x) \leq v_i/r_i \quad (i \in MI \cup ME)$  が成り立つ。

(仮定 4)

$$\max(t^2 + 2t\|A\| + \|A\|^2, 2t\|A\| + \|A\|^2 + \|r\|^{-2}) \leq \sigma^{-2}.$$

(仮定 5)

$$d^T B d \geq \delta \|d\|^2.$$

(仮定 6)

$$\|d\|^2 + \frac{1}{2} \bar{g}'^T R \bar{g} \geq (u-v)^T E (u-v).$$

(仮定 7)

$$\gamma = \sigma \min_{i \in MI \cup ME} \left\{ \frac{\delta}{2}, \frac{\varepsilon_i \delta}{2} - \frac{1}{r_i(1-0.25\delta)} \right\} > 0.$$

以上の仮定の下で, 次の式が成り立つ。

$$-\nabla \Phi(x, v; r)^T \begin{pmatrix} d \\ u-v \end{pmatrix} \geq \gamma \left\| \begin{pmatrix} d \\ u-v \end{pmatrix} \right\| \|\nabla \Phi(x, v; r)\|. \tag{4.18}$$

(注意) (4.19) 式が (仮定 6) の十分条件であり, 4.2 節で述べたベクトルペナルティ  $r$  の更新方法は, 仮定 6 の代わりに, この十分条件を満足するように構成してある。

$$\frac{1}{m}\|d\|^2 + \frac{1}{2}r_i \bar{g}_i' \bar{g}_i \geq \varepsilon_i (u_i - v_i)^2. \quad (4.19)$$

〈仮定と結論の意味〉

(仮定1) この仮定を満足するように, 4.2 節で述べたように  $\delta, \varepsilon, r$  を決めている.

(仮定2) 元々の非線形最適化問題が最適解を持つならば, 最適化が十分進み, 探索点が最適点に近づいたときには, 明らかに成り立つ.

(仮定3) 二次計画問題の解における有効制約条件と, 元々の非線形計画問題の探索点における有効制約条件が一致することを意味している. これは, 最適化が十分進み, 探索点が最適点に近づいたときには, 明らかに成り立つ.

(仮定4) 十分小さい  $\sigma > 0$  に対して, 成り立つ.

(仮定5) 行列  $B$  が正定値であれば, これを満足するように  $\delta$  を選ぶことができる. 行列  $B$  の正定値性は, 理論的には保証されている. 数値的に成り立たなくなったときには, 行列  $B$  を単位行列にリセットする. 結局, 常に仮定5は満足している.

(仮定6), (仮定7) 仮定6の十分条件である(4.19)式と仮定7を満足するために, 4.2 節で述べたように,  $\varepsilon_i$  と  $r_i$  を選ぶ.

(結論) 定理の結論である(4.18)式は, 直線探索の探索方向ベクトルと  $\nabla \Phi(x, v; r)$  の勾配の内積が負になることを示している. つまり, 直線探索の探索方向ベクトルが  $\Phi(x, v; r)$  の降下方向になることを意味している. 各反復で, これが満たされれば, 逐次二次計画法による最適化が成功することを保証できる.

〈証明〉 仮定1~仮定4から, (4.20)式を示すことができる.

$$\left\| \begin{pmatrix} d \\ u-v \end{pmatrix} \right\| \geq \sigma \|\nabla \Phi(x, v; r)\|. \quad (4.20)$$

仮定1, 仮定5~仮定7および(4.20)式から(4.18)式を示すことができる. (証明終)

証明の詳細は付録に示す.

#### 4.4 改良の効果

従来, 非線形最適化問題を解く際, 目的関数や制約条件のバランスを考えて, スケーリングと呼ばれる処理を施さなければならなかった. スケーリングには, (1)目的関数や制約条件の各々を定数倍する, (2)変数変換する等の方法があった. しかし, 非線形最適化との関係なしの適切なスケーリングは難しい. 例えば, 初期状態では良いスケーリングが反復が進むと望ましくないスケーリングになってしまうこともある.

適切なスケーリングが行われていない場合, 最適化の効率が悪くなることや失敗することがある.

ベクトルペナルティ型拡張ラグランジュ関数が, このスケーリングをある程度自動的に行うことを期待できる. 各制約条件ごとにペナルティパラメタの値が決まる. そのため, 制約条件どうしのバランスが崩れているとき, 自動的に補正できる. さらに, 目的関数とのバランスが崩れているときにも制約条件ごとにバランスを補正できる. さらに, この補正は, 非線形最適化と同時に行うので, 従来からあるスケーリングの欠点を克服できることが期待できる.

これにより, 目的関数や制約条件のバランスが崩れた問題に対する逐次二次計画法の効率や信頼性が向上することが期待できる.

ベクトルペナルティは, スカラペナルティに比べて制約条件-1個多い値を記憶し, 更新するための計算をしなければならない. しかし, 記憶容量, 計算量のどちらに関しても他の部分に必要な量に比較して問題にならない.

#### 4.5 数値実験結果

これまで述べてきた手法を基にコンピュータプログラムを作成した. 二次計画問題の解法としては, GI法<sup>9)</sup>, 最小二乗法<sup>9)</sup>をプログラム化した. また, 適応型アルミーホ法<sup>9)</sup>, 簡易型アルミーホ法<sup>10)</sup>などをプログラム化した.

このプログラムをシトコウスキイの非線形最適化問題集<sup>9)</sup>に適用した. メリット関数としてスカラペナルティ型拡張ラグランジュ関数を使った場合には, 事前のスケーリングなしには106番と116番を解くことができなかった. 一方, メリット関数としてベクトルペナルティ型拡張ラグランジュ関数を使った場合には, それらの問題を解くことができた.

図2に106番を示す.

106番は1番目の制約条件に  $-0.0025x_4$ , 4番目の制約条件に  $-833.33252x_4$  という項を含むなどにより, 互いに勾配の大きさが大きく異なることがある.

二次計画問題の解は, 二次計画問題に対応するキューン・タッカー条件を満足する. キューン・タッカー条件の中の一つに, ラグランジュ関数の勾配の大きさが0になるという条件がある. これは, 制約関数の勾配に重みを付けた線形結合が目的関数の勾配と等しくなることを要求している. 重みとはラグランジュ乗数であり, 不等号制約条件に対応するものは0以上である. 通常, 有効な(探索点が境界上にある)制約

目的関数

$$f(x) = x_1 + x_2 + x_3$$

制約条件

$$\begin{aligned} &1 - 0.0025(x_6 + x_8) \geq 0, \\ &1 - 0.0025(x_6 + x_7 - x_8) \geq 0, \\ &1 - 0.01(x_6 - x_8) \geq 0, \\ &x_1 x_2 - 833.33252 x_4 - 100 x_1 + 83333.333 \geq 0, \\ &x_2 x_7 - 1250 x_5 - x_2 x_4 + 1250 x_4 \geq 0, \\ &x_3 x_6 - 1250000 - x_3 x_5 + 2500 x_5 \geq 0, \\ &100 \leq x_1 \leq 10000, \\ &1000 \leq x_i \leq 10000 \quad (i=2,3), \\ &10 \leq x_i \leq 1000 \quad (i=4, \dots, 8). \end{aligned}$$

出発点

$$x_0 = (5000, 5000, 5000, 200, 350, 150, 225, 425),$$

$$f(x_0) = 15000.$$

最適解

$$x^* = (579.3167, 1359.943, 5110.071, 182.0174, 295.5985, 217.9799, 286.4162, 395.5979),$$

$$f(x^*) = 7049.330923.$$

図 2 シトコウスキイ問題集の 106 番<sup>\*)</sup>  
Fig. 2 Schittkowski's problem No. 106.

条件に対してのみラグランジュ乗数が 0 より大きい。結局、有効な制約条件の勾配に正の重みを付けた和のベクトルが目的関数の勾配に等しくならなければならない。この条件を満足するために、勾配の大きな制約関数に対応するラグランジュ乗数が、勾配の小さな制約関数に対応するラグランジュ乗数に比べて小さくなる。

106 番の最適化では、1 番目の制約条件の勾配の大きさが 4 番目の制約条件の勾配の大きさに比べて小さいため、1 番目の制約条件に対応するラグランジュ乗数の値が大きくなる。(3.3) 式よりラグランジュ乗数が大きくなると  $\epsilon$  が小さくなり、(3.4)、(3.5) 式より、 $\epsilon$  が小さいとペナルティパラメータ  $r$  が大きくなる。つまり、1 番目の制約条件によってペナルティパラメータが大きくなる。ところが、そのような大きなペナルティパラメータは 4 番目の制約条件のように勾配の大きい制約条件の境界の状況を厳しくする。4 番目の制約条件を満足しない領域への探索を全く許さなくなる。その厳しさが探索点の改良を妨げる。直線探索時に、ほとんど探索点の改良が行えないという Maratos effect と同様の現象が起きる。

116 番も同様の問題であり、同様の現象が起きる。この現象は、われわれの作成したコンピュータプログラム固有の問題ではなく、その形からスカラペナルティ型拡張ラグランジュ関数が一般に持っている数値計算上の欠点だと考えることができる。スカラペナルテ

表 1 シトコウスキイ問題集 106 番の結果

Table 1 Summary of numerical experiments for Schittkowski's problem No. 106.

項番	二次計画法	$\phi$	$f(x)$	CPU (s) 反復	$\ FL\ _{r^*}$	関数計算回数
1	G I	スカラ	1.03D4	2.23 1001	1.7D0 4.3D-11	F 1001 DF 15 G 1001 DG 15
2	G I	ベクトル	7.05D3	1.10 364	1.6D-5 7.1D-4	F 364 DF 359 G 364 DG 359
3	L S	スカラ	1.03D4	2.24 1001	1.7D0 0.0	F 1001 DF 15 G 1001 DG 15
4	L S	ベクトル	7.05D3	1.37 364	1.6D-5 7.1D-4	F 364 DF 359 G 364 DG 359

G I : G I 法

L S : 最小二乗法

スカラ : スカラペナルティ型拡張ラグランジュ関数

ベクトル : ベクトルペナルティ型拡張ラグランジュ関数

$r^*$  : 制約条件の不満足分の合計

F : 目的関数計算回数

G : 制約関数計算回数

DF : 目的関数導関数計算回数

DG : 制約関数導関数計算回数

表 2 シトコウスキイ問題集 116 番の結果

Table 2 Summary of numerical experiments for Schittkowski's problem No. 116.

項番	二次計画法	$\phi$	$f(x)$	CPU (s) 反復	$\ FL\ _{r^*}$	関数計算回数
1	G I	スカラ	1.70D2	4.25 1001	1.2D0 2.0D-5	F 7383 DF 838 G 7383 DG 838
2	G I	ベクトル	9.76D1	1.45 194	1.3D-5 3.4D-12	F 987 DF 193 G 987 DG 193

ィ型拡張ラグランジュ関数が数値計算上の厳しい状況に対応できなくなっている。

ベクトルペナルティ型拡張ラグランジュ関数を使えば、上記の現象を避けることができると考えられる。

スカラペナルティ型拡張ラグランジュ関数を使うと

解けなかった問題を、ベクトルペナルティ型拡張ラグランジュ関数を使えば解けたことをここに示す。問題番号 106 番および 116 番に関して各手法の結果を表 1～表 2 にまとめる。数値実験は HITAC S-820 モデル 80 上の倍精度計算で行った。

例えば、106 番を解いたときには、最終的にスカラペナルティ  $r=10^8$  であったのに対して、ベクトルペナルティは次のとおりであった。

$$r_1 \sim r_3 \doteq 10^{14},$$

$$r_4 \sim r_6 \doteq 10^3,$$

$$r_7 \sim r_{22} = 2.$$

シトコウスキイの問題集の中の他の問題のうち、制約条件が二つ以上ない問題に関しては、二つの関数の形から明らかなように、二つの関数の結果は全く同じであった。また、119 番、118 番、117 番、113 番、112 番のように制約条件の勾配の大きさがほぼ同じである問題に関しては、二つの関数の結果は同様であった。逆に、114 番のように制約条件の勾配の大きさが互いに異なる問題に関しては 106 番や 116 番と同様にベクトルペナルティ型拡張ラグランジュ関数の結果が優れていた。

## 5. む す び

シトコウスキイの提案した逐次二次計画法の中でメリット関数として用いられている拡張ラグランジュ関数の改良を提案した。ペナルティパラメタをベクトルとし、各制約条件に対応させたベクトルペナルティ型拡張ラグランジュ関数である。この関数を逐次二次計画法のために使用するために必要なペナルティパラメタの更新方法を提案し、収束を保証するための定理を証明した。

ベクトルペナルティ型拡張ラグランジュ関数は、目的関数や制約条件のバランスが崩れた問題に対してスカラペナルティ型拡張ラグランジュ関数が有効でなくなることに對する対策である。実際、数値実験によって互いに勾配の大きさが大きく異なる制約条件をもつ問題に有効であることを確認できた。

ベクトルペナルティ型拡張ラグランジュ関数は、逐次二次計画法以外の多くの方法に対しても適用可能であると考えられる。本論文と同様の定理を証明し、有効性を確認することが今後の課題である。

**謝辞** 本論文のための研究に関しては、池辺八洲彦教授（筑波大学）、小柳義夫教授（東京大学）、今野浩教授（東京工業大学）、杉原正顯助教授（東京大学）を

はじめとして多くの方々の御協力をいただきましたことに深謝いたします。

## 参 考 文 献

- 1) Schittkowski, K.: The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function—part 1: Convergence Analysis, *Numerische Mathematik*, Vol. 38, pp. 83-114 (1981).
- 2) Hock, W. and Schittkowski, K.: *Test Examples for Nonlinear Programming Codes*, Springer-Verlag (1980).
- 3) Yamashita, H.: A Globally Convergent Constrained Quasi-Newton Method with an Augmented Lagrangian Type Penalty Function, *Math. Prog.*, Vol. 23, pp. 75-86 (1982).
- 4) Goldfarb, D. and Idnani, A.: A Numerically Stable Dual Method for Solving Strictly Convex Quadratic Programs, *Math. Prog.*, Vol. 27, pp. 1-33 (1983).
- 5) Schittkowski, K.: The Nonlinear Programming Method of Wilson, Han, and Powell with an Augmented Lagrangian Type Line Search Function—part 2: An Efficient Implementation with Linear Least Squares Subproblems, *Numerische Mathematik*, Vol. 38, pp. 115-127 (1981).
- 6) Tamura, M. and Kobayashi, Y.: Application of Sequential Quadratic Programming Software to an Actual Problem, *Math. Prog.*, Vol. 52, pp. 19-27 (1991).
- 7) Han, S. P.: Superlinearly Convergent Variable Metric Algorithms for General Nonlinear Programming Problems, *Math. Prog.*, Vol. 11, pp. 263-282 (1976).
- 8) Powell, M. J. D.: Convergence Properties of Algorithms for Nonlinear Optimization, *SIAM Rev.*, Vol. 28, No. 4, pp. 487-500 (1986).
- 9) 尾田十八, 木戸雄二: FRP 構造における繊維分布の最適化手法について, 日本機械学会論文集 (A 編), Vol. 49, No. 441, pp. 547-555 (1983).
- 10) Vanderplaats, G. N. and Salajegheh, E.: An Efficient Approximation Technique for Frequency Constraints in Frame Optimization, *International Journal for Numerical Methods*, Vol. 26, pp. 1057-1069 (1988).
- 11) Kodiyalam, S. and Vanderplaats, G. N.: Shape Optimization of Three Dimensional Continuum Structures via Force Approximation Technique, *AIAA Journal*, Vol. 27, No. 9, pp. 1256-1263 (1989).

付 録

〈記法〉

$$A_J: A \text{ の } i \in J \text{ 列を集めた行列.} \quad (\text{A. 1})$$

$$\bar{g}'(x) = (r_1 \bar{g}_1'(x), \dots, r_m \bar{g}_m'(x))^T. \quad (\text{A. 2})$$

〈定理の証明〉

まず, (4.20)式を証明する. ただし, 関数の引数を示す  $(x)$  は省略する.

二次計画問題に関するキューン・タッカー条件より, 次の式が成り立つ.

$$Bd + \nabla f - Au = 0. \quad (\text{A. 3})$$

また,  $w = g + A^T d$  とおくと以下の式が成り立つ.

$$w \geq 0. \quad (\text{A. 4})$$

$$w^T u = 0. \quad (\text{A. 5})$$

$$u_i \geq 0 \quad (i \in MI). \quad (\text{A. 6})$$

以上のことから, (4.20)式は, 次のように証明できる.

$$\begin{aligned} & \|\nabla \Phi(x, v; r)\|^2 \\ &= \|\nabla f - A(\bar{v} - \bar{g})\|^2 + \|\bar{g}'\|^2 \\ &= \|-Bd + A(u - \bar{v}) + AR\bar{g}\|^2 \\ &\quad + \sum_{i \in J} g_i^2 + \sum_{i \in K} \frac{v_i^2}{r_i^2} \\ &\leq (\|B\| \|d\| + \|A\| \|u - \bar{v}\| + \|A\| \|R\bar{g}\|)^2 \\ &\quad + \|A_J d\|^2 + \sum_{i \in K} \frac{(u_i - v_i)^2}{r_i^2} \\ &\leq (\|B\| \|d\| + \|A\| \|u - v\| + \|A\| \|r\| \|A_J^T d\|)^2 \\ &\quad + \|A\|^2 \|d\|^2 + \frac{\|u - v\|^2}{\|r\|^2} \\ &\leq \{(\|B\| + \|r\| \|A\|^2) \|d\| + \|A\| \|u - v\|\}^2 \\ &\quad + \|A\|^2 \|d\|^2 + \frac{\|u - v\|^2}{\|r\|^2} \\ &\begin{cases} \cdot \|d\| < \|u - v\| \text{ ならば} \\ \leq (t^2 + \|A\|^2) \|d\|^2 \\ \quad + \left(2t\|A\| + \|A\|^2 + \frac{1}{\|r\|^2}\right) \|u - v\|^2 \\ \cdot \|d\| \geq \|u - v\| \text{ ならば} \\ \leq (t^2 + 2t\|A\| + \|A\|^2) \|d\|^2 \\ \quad + \left(\|A\|^2 + \frac{1}{\|r\|^2}\right) \|u - v\|^2 \end{cases} \\ &\leq (\|d\|^2 + \|u - v\|^2) / \sigma^2 \\ &= \frac{1}{\sigma^2} \left\| \begin{pmatrix} d \\ u - v \end{pmatrix} \right\|^2. \quad (\text{A. 7}) \end{aligned}$$

次に, (4.18)式を証明する.

$$-\nabla \Phi(x, v; r)^T \begin{pmatrix} d \\ u - v \end{pmatrix}$$

$$\begin{aligned} &= -\nabla f^T d + (\bar{v} - \bar{g})^T A^T d + \bar{g}'^T (u - v) \\ &= d^T B d - u^T A^T d + \bar{v}^T A^T d - \bar{g}^T A^T d + \bar{g}'^T (u - v) \\ &= d^T B d - (u - \bar{v})^T A^T d - \bar{g}^T w + \bar{g}'^T g + \bar{g}'^T (u - v) \\ &= d^T B d - (u - \bar{v})^T w + (u - \bar{v})^T g \\ &\quad - \bar{g}^T w + \bar{g}'^T g + \bar{g}'^T (u - v) \\ &= d^T B d + (\bar{v} - \bar{g})^T w + \bar{g}'^T \bar{g} + 2\bar{g}'^T (u - v) \\ &\quad + \sum_{i \in K} \left( g_i u_i - \frac{v_i}{r_i} u_i + \frac{v_i^2}{r_i} \right) \\ &\geq \delta d^T d + \bar{g}'^T \bar{g} + 2\bar{g}'^T (u - v) \\ &\quad + \sum_{i \in K} \frac{v_i^2}{r_i} + \sum_{i \in K} u_i \left( g_i - \frac{v_i}{r_i} \right) \\ &\geq \delta d^T d + \bar{g}'^T \bar{g}' + 2\bar{g}'^T (u - v) \\ &= \frac{\delta}{2} d^T d + \frac{\delta}{2} d^T d + \frac{\delta}{4} \bar{g}'^T \bar{g}' \\ &\quad + \left(1 - \frac{\delta}{4}\right) \bar{g}'^T \bar{g} + 2\bar{g}'^T (u - v) \\ &\geq \frac{\delta}{2} d^T d + \frac{\delta}{2} (u - v)^T E (u - v) \\ &\quad + \left\| \sqrt{1 - \frac{\delta}{4}} R^{1/2} \bar{g}' + \left( \sqrt{1 - \frac{\delta}{4}} R^{1/2} \right)^{-1} (u - v) \right\|^2 \\ &\quad - \left(1 - \frac{\delta}{4}\right)^{-1} (u - v)^T R^{-1} (u - v) \\ &\geq \frac{\delta}{2} \|d\|^2 + \frac{\delta}{2} (u - v)^T E (u - v) \\ &\quad - \left(1 - \frac{\delta}{4}\right)^{-1} (u - v)^T R^{-1} (u - v) \\ &\geq \frac{\delta}{2} \|d\|^2 + \sum_{i \in MI \cup ME} \left\{ \frac{\delta}{2} \varepsilon_i - \frac{1}{\left(1 - \frac{\delta}{4}\right) r_i} \right\} (u_i - v_i)^2 \\ &\geq \frac{\gamma}{\sigma} \left\| \begin{pmatrix} d \\ u - v \end{pmatrix} \right\|^2 \\ &\geq \gamma \left\| \begin{pmatrix} d \\ u - v \end{pmatrix} \right\| \|\nabla \Phi(x, v; r)\|. \quad (\text{A. 8}) \end{aligned}$$

(証明終)

(平成2年7月11日受付)  
(平成3年4月9日採録)





**田村 正義 (正会員)**

昭和 37 年生. 昭和 59 年筑波大学第 3 学群情報学類卒業. 昭和 61 年同大学大学院工学研究科修士課程修了. 同年(株)日立製作所入社. 現在, 同社ソフトウェア開発本部にて, 非線形最適化プログラムの開発に従事. 日本応用数理学会, 日本 OR 学会会員.



**小林 康弘 (正会員)**

昭和 22 年 10 月 21 日生. 昭和 45 年 3 月東京大学工学部原子力工学科卒業. 昭和 50 年 3 月同大学院原子力工学専攻博士課程修了. 同年 4 月(株)日立製作所入社. 昭和 53 年 4 月より同社エネルギー研究所勤務, 現在に至る. 原子力プラントの信頼性・安全性, 新型 BWR 炉心の概念設計, 省エネルギー面からのシステム評価, プラント設計自動化の研究開発に従事. 工学博士. 人工知能学会, 電気学会, 日本原子力学会, IEEE, AAAI などの会員.



**野中 久典 (正会員)**

昭和 37 年生. 昭和 59 年大阪大学工学部原子力工学科卒業. 昭和 61 年同大学院修士課程修了. 同年(株)日立製作所エネルギー研究所入社. 知識工学的手法を用いるプラント建設工程計画支援システムの開発, 非線形最適化システムの開発に従事. 日本原子力学会会員.