日本オペレーションズ・リサーチ学会論文誌
Journal of the Operations Research
Society of Japan
Vol. 19, No. 4, December, 1976

一変数多峰性関数に対する最適値探索法の研究

青山学院大学　正道寺　勉

AN ALGORITHM FOR OBTAINING A GLOBAL OPTIMUM FOR ONE VARIABLE MULTI-MODAL FUNCTIONS

TSUTOMU SHOHDOHJI, *Aoyama Gakuin University*

(Received August 25, 1975; Revised April 13, 1976)

Abstract. This paper attempts to explore the algorithm of a new method for solving non-linear optimization problems. The study focuses on the method for obtaining maximum or minimum value for one variable multi-modal function. It is shown that the algorithm developed here can be applied to solving the problem to optimize any objective function defined on a bounded closed interval which satisfies the Lipschitz's condition.

The traditional optimizing methods deal primarily with the uni-modal objective function for obtaining optimum value. This paper studies the method to obtain the global solution for multi-modal objective function. This algorithm, through using the Lipschitz's constant, can determine the upper limit of the obtained solution and evaluate the relative error to the true optimum value. This algorithm makes use of the relative error stopping rule. The accuracy of the solution, therefore, does not depend on the form of the objective function.

This method can also be applied to some of the separable objective functions in the fixed charge problem. Numerical tests with the algorithm produced fairly satisfactory results.

1. はじめに

現在のところ，非線型最適化手法[1)]は単峰性（unimodality）の目的関数にのみその適用が限られているので，多峰性（multimodality）の目的関数になれば大域的（global）な最大値または最小値に必ずしも到達するとは限らない。一般に一変数，多変数を問わず従来の最適化手法は，得られた解と真の解との誤差がどれ位であるかの保証がないので，真の解の上限または下限を定めることができないという欠点がある。それでは実際問題への適用が充分でないと思われる。

多変数多峰性関数に関する研究は，Branin, F.H., Jr.〔2〕,〔3〕の微分方程式を解くことにより大域解を得ようとする方法や津田・佐藤〔12〕によってなされたランダム・サンプリングによる方法[2)]などが挙げられる。しかしながら，現在のところ決定論的（deterministic）な立場での研究はあまりなされていないようである。

一方，一変数多峰性関数に関する研究は，Brent，R.P.〔4〕に紹介されているが，本論文はこれとは本質的に異なった理論によるものであり，竹内〔11〕の提案した方法をさらに発展させたものである。特に，一変数多峰性関数の最適化問題に対して，最大値を探す有効なアルゴリズムおよび数値実験結果を示す。

本アルゴリズムは，有界閉区間で定義される目的関数をリプシッツ条件（Lipschitz's condition）の下で最大または最小にする場合に適用するが，その応用範囲は広いものと思われる。また本アルゴリズムは，リプシッツ定数（Lipschitz's constant）を使用することにより目的関数が少なくとも探索区間内では単峰性が保証されていなければならないという従来の一変数探索法の致命的欠点を取り除くことを第一の目的としている。その上，本アルゴリズムでは，従来のアルゴリズムのような区間幅によるStopping Ruleを定めずに，真の解と得られた最良解[3)]との最大相対誤差によるStopping Ruleを定めてあるので，アルゴリズムが収束して得られた最良解の精度が目的関数の関数値に左右されることがない[4)]。目的関数の種類によっては，いくら最良解の存在する区間幅を縮めたところで真の解との誤差が縮まるとは限らないことがある。実際問題では，目的関数の目標値との誤差がどれ位であるという情報の方がはるかに重要であることが多い。

本アルゴリズムの特徴は，

1） 得られた最良解は，大域的な最良解であることが理論的に保証されている。

2） 得られた最良解と真の解との誤差評価が可能である。

という二点であるが，数値実験の結果も良好である。

本論文のアルゴリズムは，目的関数が一個の独立変数で定義され，且つリプシッツ条件を満足するような連続関数を制約条件なし（ただし，独立変数の定義域に関する制約は制約条件とは見なさないものとする。）の下で最大化する場合に適用する。目的関数が多変数関数の場合には，ある条件を満足すれば，本アルゴリズムの適用が可能であることを第7章にて示す。

2. 問題設定

2・1. 最適化問題の定式化

一般にシステムの最適化問題は，①制約条件付最適化問題と②制約条件なし最適化問題とに大別できるが，この二つを定式化すれば（問題1）および（問題2）のようになる。

（問題1） 最大化 $f(X)$，$X=(x_1, x_2, \cdots\cdots, x_n)$

制約条件 $h_i(X)=0$，$g_j(X)\leqq 0$，

$(i=1, 2, \cdots\cdots, m)$，$(j=1, 2, \cdots\cdots, p)$

$X \subset E^n$，$m \leqq n$

ここで，関数f，h_i，g_jは連続な実数値関数である。またE^nはn次元ユークリッド空間を示す。これらの記号は（問題2)でも同じ意味で使用する。

（問題2） 最大化 $f(X)$，$X=(x_1, x_2, \cdots\cdots, x_n)$

制約条件 $X \subset E^n$

制約条件付最適化問題は，ある種の変換[5]により，制約条件なし最適化問題に変換することが可能である。その上，制約条件なし最適化問題を解く手法は，制約条件付最適化問題を解く手法として利用されることが多い。

2・2 最小化と最大化

最小化問題，例えば $\min_{X \subset E^n} f(X)$ は $\max_{X \subset E^n} \{-f(X)\}$ を用いることにより，容易に最大化問題に変形することができる。ここで注意しなければならないことは，最大化問題に変形して得られた最大値を元の最小化問題にあてはめる場合には，符号を反転させねばならない。すなわち，$\min_{x \subset E^n} f(x) = -\max_{x \subset E^n} \{-f(x)\}$ である。以上のことを考慮して，本論文で扱う最適化問は，目的関数における変数の定義域に関する制約条件は別として，制約条件なしの最適化問題に対し，目的関数の最大値を求める場合のみを考えることにする。

2・3 一変数最適化問題

現在の一変数最適化手法は，目的関数が少なくとも探索区間内において，単峰であることを大前提においている。それ故，多くの探索点（または試行点）を用いてそれらの探索点に対応する関数値を比較する一様探索法で得られた解には大域的な最良解であるという保証は与えられない。一般に黄金分割探索法，フィボナッチ探索法，ニュートン法などを多峰性関数の最適化問題に適用したならば，大域的な最適解が得られないことがある。本論文では，目的関数が多峰性関数の場合に有効なアルゴリズムの提出を試みる。

3. アルゴリズム

3・1 概 要

与えられた実数値連続関数 $f \in C^1[a, b]$ を，制約条件なしの下で最大化することを考える。このとき，関数 f にのみ依存する定数 L が存在して，任意の x_1，$x_2 \in [a, b]$ に関して，

$$(3 \cdot 1) \quad |f(x_2) - f(x_1)| \leqq L \cdot |x_2 - x_1|$$

が成立するものとする。ここで L は，リプシッツ定数と呼ばれる定数である。

関数 $f(x)$ を閉区間 $[a, b]$ 内で探索する場合には，定数 L を決定せねばならない。リプシッツ定数 L の決定の仕方については第6章で述べる。

探索区間内における点列を $\{x_i\}_{i=0}^{i=n}$ とすれば，点列 $\{x_i\}_{i=0}^{i=n}$ は，

$$(3 \cdot 2) \quad x_i = a + i \times h, \quad (i = 0, 1, \cdots\cdots, n)$$

で表現できる。ここで，h は閉区間 $[a, b]$ を n 等分した値である。すなわち，

$$(3 \cdot 3) \quad h = \frac{b-a}{n}$$

であり，$a = x_0$，$b = x_n$ となる。

多峰性関数の最適化に際して最も重要なことは，真の最適値（大域的最適値）を見逃がさないことである。

閉区間 $[a, b]$ が n 等分されたそれぞれの区間内で，不等式：

(3・4)　$|f(x_{i+1})-f(x_i)| \leqq L \cdot |x_{i+1}-x_i|$，$(i=0, 1, \cdots\cdots, n)$

が成立する。不等式(3・4)は，閉区間$[x_i, x_{i+1}]$$(i=0, 1, \cdots\cdots, n)$での最大値が$L \cdot |x_{i+1}-x_i|$で定められていて，

(3・5)　$\min\{f(x_{i+1}), f(x_i)\}$[6]$+L \cdot |x_{i+1}-x_i|$，

$(i=0, 1, \cdots\cdots, n)$

を決して超えないことを示している。

(3・2)式より容易に，$|x_{i+1}-x_i|=h$，$(i=0, 1, \cdots\cdots, n)$であることがわかる。

ここで，

(3・6)　$y_i=f(x_i)$，$(i=0, 1, \cdots\cdots, n)$

とし，y_0，y_1，……，y_nの中での最大値を$\bar{y}$とすれば，

(3・7)　$\bar{y} \leqq \max_{X \in [a,b]} f(x) \leqq \max_i \{\min(y_{i+1}, y_i)+L \times h\}$，

$(i=0, 1, \cdots\cdots, n)$

(3・8)　$\leqq \bar{y}+L \times h$

が成立する。この証明は第4章にて示す。

関数$f(x)$の真の最大値は，(3・5)式より，

(3・9)　$\bar{y} \leqq \min_i(y_{i+1}, y_i)+L \times h$，$(i=0, 1, \cdots\cdots, n)$

を満足するようなiに対する閉区間$[x_i, x_{i+1}]$の中に必らず存在することがわかる。探索が初期の場合には，不等式(3・9)の成立する閉区間は複数個存在する。探索が進むにつれ，これらの閉区間のみについて再びn等分すればよい[7]。

最終的には，真の最大値が存在する変数の閉区間および真の最大値が必らず含まれる関数値の範囲(すなわち，下限値αと上限値βを$[\alpha, \beta]$のように示す。)を得る。これらの数値は，それぞれ不等式(3・9)と(3・7)から計算される。

3・2　計算手順

リプシッツ定数Lによる最適値探索アルゴリズムは，次のようになる。

ステップ1：初期値設定および予備計算

a，b，L，nの値を入力する。

探索区間$[a, b]$をn等分し，それを区間幅hと定める。

ステップ2：探索点x_iとそのときの関数値y_iをそれぞれ$x_i=a+i \times h$，$y_i=f(x_i)$，$(i=0, 1, \cdots\cdots, n)$として計算する。

ステップ3：関数値y_iの中での最大値YSTARを求める。

YSTAR$=\max\{f(x_i^{(j)})\}$[8]，$(i=0, 1, \cdots\cdots, n)$

ステップ4：収束の判定

収束の判定条件は，得られた最良値による相対誤差により定めており，相対誤差が10^{-3}以下にな

ればアルゴリズムは終了する。収束の判定条件を満たさなければステップ5に進む。

ステップ5：最大値の推定

真の最大値を下まわらない上限値YMAXを求める。

$$YMAX=\min\{y_i^{(j)},\ y_{i+1}^{(j)}\}+L\times h,$$

$(i=0,\ 1,\ \cdots\cdots,\ n)$

次にYSTARとYMAXとを比較し，YSTAR ≦YMAXならば，$YGREAT=\max_i(YMAX)$ を計算する。

ステップ6：探索区間の改善

はじめの探索区間〔a, b〕より小さな区間〔x_{i-1}^*, x_i^*〕[9]が定まる。これは通常複数個存在する。これを再びn等分する。そのときの区間幅hを，$h\leftarrow h/n$，また$n\leftarrow n+$(YSTAR≦YMAXを満たす回数)として，ステップ2へもどる。

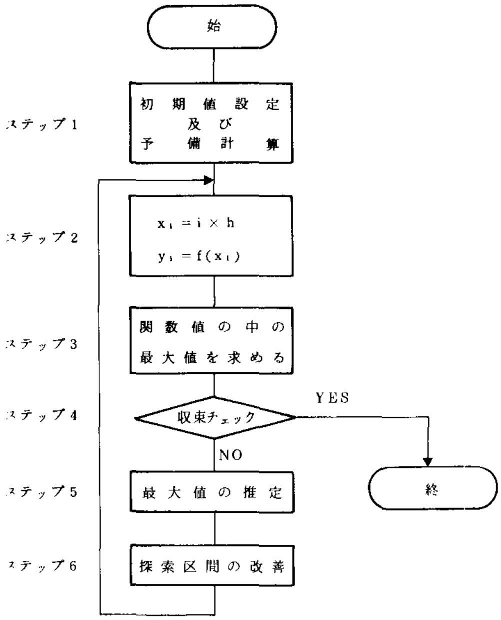

図1　探索アルゴリズム

4. アルゴリズムの理論的考察

一変数関数f(x)に関して，(3・6)式で決まる関数値を考え，1回目の探索における関数f(x)の評価値をそれぞれ$y_0^{(1)}$, $y_1^{(1)}$, $\cdots\cdots$, $y_{m1}^{(1)}$とする。以下同様にして2回目以降は，$y_0^{(\ell)}$, $y_1^{(\ell)}$, $\cdots\cdots$, $y_{m\ell}^{(\ell)}$, $(\ell=2,\ 3,\ \cdots\cdots,\ k)$となる。ただし，$m_\ell\leqq n$, $(\ell=1,\ 2,\cdots\cdots,\ k)$である。

ここで，閉区間〔a, b〕における関数f(x)の最大値を求めることを考える。まず記号を次のように定める。

定義1：

$$(4\cdot1)\quad \bar{y}^{(\alpha)}=\max_{0\leqq\ell\leqq m_k} y_\ell^{(\alpha)},\ (\alpha=1,\ 2,\ \cdots\cdots,\ k)$$

$$(4\cdot2)\quad \hat{y}=\max_{a\leqq x\leqq b} f(x)$$

$$(4\cdot3)\quad h^{(\alpha)}=\left|x_{i+1}^{(\alpha)}-x_i^{(\alpha)}\right|,\ (i=0,\ 1,\ \cdots\cdots,\ n),\ (\alpha=1,\ 2,\ \cdots\cdots,\ k)$$

とおき，不等式(3・4)のリプシッツ条件の下では，次の定理1が成立する。

定理1：

$$(4\cdot4)\quad \bar{y}^{(\alpha)}\leqq\hat{y}\leqq\max_i\{\min(y_i^{(\alpha)},\ y_{i+1}^{(\alpha)})+L\times h^{(\alpha)}\},\ (i=0,\ 1,\ \cdots\cdots,m_k),$$

$(\alpha=1,\ 2,\ \cdots\cdots,\ k)$

$$(4\cdot5)\quad \leqq\bar{y}^{(\alpha)}+L\times h^{(\alpha)},\ (\alpha=1,\ 2,\ \cdots\cdots,\ k)$$

証明：

定義1を用いて不等式（3・4）を書き替えれば，

（4・6） $|y^{(\alpha)}_{i+1} - y^{(\alpha)}_{i}| \leqq L\,|x^{(\alpha)}_{i+1} - x^{(\alpha)}_{i}|$，（$i=0, 1, \cdots\cdots, n$），

（$\alpha=1, 2, \cdots\cdots, k$）

となる。ゆえに，

（4・7） $\hat{y} \leqq \min_{0 \leqq i \leqq m_k} (y^{(\alpha)}_{i}, y^{(\alpha)}_{i+1}) + L\,|x^{(\alpha)}_{i+1} - x^{(\alpha)}_{i}|$，（$\alpha=1, 2, \cdots\cdots, k$）

（4・8） $\leqq \max_{0 \leqq i \leqq m_k} \{\min(y^{(\alpha)}_{i}, y^{(\alpha)}_{i+1}) + L \times h^{(\alpha)}\}$，（$\alpha=1, 2, \cdots\cdots, k$）

（4・9） $\leqq \max_{0 \leqq i \leqq m_k} (y^{(\alpha)}_{i}) + L \times h^{(\alpha)}$，（$\alpha=1, 2, \cdots\cdots, k$）

（4・10） $= \bar{y}^{(\alpha)} + L \times h^{(\alpha)}$，（$\alpha=1, 2, \cdots\cdots, k$）

Q．E．D．

よって，定理1により得られた最大値（推定値）は，真の最大値を超えないことが示された。

5. 誤差解析

3・2のアルゴリズムにおいて，第kステップで探索が完了したものとする。

定理1より，真の最大値 $\hat{y}$，すなわち，

（5・1） $\hat{y} = f(\hat{x})$

を与える点 $\hat{x}$ は，

（5・2） $\hat{y}^{(k)} \leqq \min_{0 \leqq i \leqq m_k} (y^{(k)}_{i}, y^{(k)}_{i+1}) + L \times h^{(k)}$

となる閉区間〔x_r，x_{r+1}〕内に必ず存在する。ただし，rは第kステップで不等式（5・2）を満足するiの値である。

次に，得られた解の誤差解析を行なう。

（4・4）式で $\alpha=k$ と固定し，辺々から $\bar{y}^{(k)}$ を引けば，

（5・3） $0 \leqq \hat{y} - \bar{y}^{(k)} \leqq \max_{0 \leqq i \leqq m_k} \{\min(y^{(k)}_{i}, y^{(k)}_{i+1}) + L \times h^{(k)}\} - \bar{y}^{(k)}$

となる。相対誤差は（5・3）式より，

（5・4） $$0 \leqq \frac{\hat{y} - \bar{y}^{(k)}}{\hat{y}} \leqq \frac{\max_{0 \leqq i \leqq m_k} \{\min(y^{(k)}_{i}, y^{(k)}_{i+1}) + L \times h^{(k)}\} - \bar{y}^{(k)}}{\hat{y}}$$

となるが，このままでは不等式（5・4）の右辺の値は定まらない。定理1より，$\bar{y}^{(k)} \leqq \hat{y}$ であるから，これを不等式（5・4）の右辺の項に適用すれば，

（5・5） $$\frac{\max_{0 \leqq i \leqq m_k} \{\min(\bar{y}^{(k)}_{i}, y^{(k)}_{i+1}) + L \times h^{(k)}\} - \bar{y}^{(k)}}{\hat{y}}$$

$$\leqq \frac{\displaystyle\max_{0\leqq i\leqq m_k} \{\min(y_i^{(k)},\ y_{i+1}^{(k)}) + L\times h^{(k)}\} - \bar{y}^{(k)}}{\bar{y}^{(k)}}$$

が成立する。この不等式（5・5）の両辺は計算可能である。よって相対誤差は,

$$(5\cdot 6)\qquad 0\leqq \frac{|y-\bar{y}^{(k)}|}{|\hat{y}|}\leqq \frac{\left|\displaystyle\max_{0\leqq i\leqq m_k} \{\min(y_i^{(k)},\ y_{i+1}^{(k)}) + L\times h^{(k)}\} - \bar{y}^{(k)}\right|}{|\bar{y}^{(k)}|}$$

のようになる。第3章で述べたアルゴリズムでは，不等式（5・6）を収束判定条件に使っている。すなわち，第kステップで探索が完了したので,

$$(5\cdot 7)\qquad \left|\frac{\displaystyle\max_{0\leqq i\leqq m_k} \{\min(y_i^{(k)},\ y_{i+1}^{(k)}) + L\times h^{(k)}\} - \bar{y}^{(k)}}{\bar{y}^{(k)}}\right| \leqq 10^{-3}$$

を満足している。

目的関数の最適値が大きな場合，または小さな場合には本アルゴリズムの収束判定条件では，収束回数が大幅に異なると予想されるけれども，目的関数を変形させることにより解決する[10]。

6. リプシッツ定数の決め方

本アルゴリズムにおいて重要なリプシッツ定数の決定について述べる。

最適化問題：

最大化　$y=f(x)$，$f(x)\in C^1[a,\ b]$

ただし，$f(x)$はリプシッツ条件を満足する関数に限る。この問題を解くにあたって，リプシッツ条件を使用するのであるが，それに付随するリプシッツ定数Lは不等式（3・4）より,

$$(6\cdot 1)\qquad \frac{|f(x_{i+1})-f(x_i)|}{|x_{i+1}-x_i|}\leqq L,\quad (i=0,\ 1,\ \cdots\cdots,\ n)$$

となり，定数Lを決定する目安になる。

リプシッツ定数Lは，不等式（6・1）の左辺に平均値の定理を適用すれば求まる。すなわち,

$$L\geqq \frac{|f(x_{i+1})-f(x_i)|}{|x_{i+1}-x_i|} = \left|\frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i}\right|,$$

$$(i=0,\ 1,\ \cdots\cdots,\ n)$$

$$(6\cdot 2)\qquad = |f'\{x_i+\theta(x_{i+1}-x_i)\}|,\quad 0<\theta<1,\ (i=0,\ 1,\ \cdots\cdots,\ n)$$

$$(6\cdot 3)\qquad \geqq \max_{x_i\in[a,\ b]} |f'(x_i)|,\quad (i=0,\ 1,\ \cdots\cdots,\ n)$$

となるから，リプシッツ定数Lは,

$$(6\cdot 4)\qquad \max_{x\in[a,\ b]} |f'(x)|\leqq L<\infty$$

を満足する数であればよいことが分かる（関数$f(x)$が必らずしも微分可能ではない連続関数であるとき，（6・4）に対応する評価式を作ることは容易である。）。

本アルゴリズムの効率の良さは，リプシッツ定数Lの選択にかかっている。すなわち，リプシッツ

定数Lが“best possible”に近ければ近い程収束が速くなる。しかし，表2を見れば分かる通り，リプシッツ定数が少々過大に決められたとしても，我々のアルゴリズムの収速性に本質的な影響はあらわれない。そして，必らずしもbest possibleではない一つのリプシッツ定数は，次の例題で示す通り比較的容易に求められる[11]。

〔例題〕： $f(x)=e^{-x}\cdot \sin x,\ x\in[0,\ 16]$

に関するリプシッツ定数Lは，次の通りである。

$$|f'(x)| = |e^{-x}(\cos x-\sin x)| \leqq |e^{-x}|\cdot|\cos x-\sin x|$$

$$\leqq |e^{-x}|\cdot(|\cos x|+|\sin x|) \tag{6・5}$$

となる。探索区間 $x\in[0,\ 16]$ を考慮に入れて不等式（6・4）及び（6・5）からリプシッツ定数Lは，

$$\begin{aligned} \max_{X\in[0,16]} |f'(x)| &\leqq \max_{X\in[0,16]} \{|e^{-x}|\cdot(|\cos x|+|\sin x|)\} \\ &\leqq \max_{X\in[0,16]} |e^{-x}|\,(\max_{X\in[0,16]} |\cos x| \\ &+ \max_{X\in[0,16]} |\sin x|) \leqq L \end{aligned} \tag{6・6}$$

となる。ここで各項のとり得る最大値を考えてみれば，

$$\max_{X\in[0,16]} |e^{-x}| = |e^{0}| = 1$$

$$\max_{X\in[0,16]} |\sin x| = \max_{X\in[0,16]} |\cos x| = 1$$

となり，これらを不等式（6・6）に代入すれば，リプシッツ定数Lは，$L\geqq 1\cdot(1+1)=2$と求められる。

7. 多変数最適化問題への応用

最適化問題の目的関数にはいろいろな種類があり，Fixed Charge問題[12]に見られるような分離型目的関数（Separable Objective Function）もある。この分離型目的関数は，

$$Z=F(x_1,\ x_2,\ \cdots\cdots,\ x_n) \tag{7・1}$$

$$=\sum_{i=1}^{n} f_i(x_i),\quad x_i\in E^n\quad (i=1,\ 2,\ \cdots\cdots,\ n) \tag{7・2}$$

のように書けるものを指す。ここで，目的関数Zの最大値及び最大値を与えるような $\bar{x}_1$，$\bar{x}_2$，… $\bar{x}_n$ を求める問題を考える。

（7・2）式に関して，第3章で述べた手法を用いれば，一変数関数 $f_1(x_1)$，$f_2(x_2)$，……，$f_n(x_n)$ のそれぞれについて，最大値を容易に求めることができる。それぞれの関数 f_i について最大値が求まったならば，目的関数Zの最大値は，それらの最大値の和として現わせる。すなわち，変数 x_1，x_2，……，x_n が各々独立であると仮定すれば，

$$\max Z=\max_{x_i}\{\sum_{i=1}^{n} f_i(x_i)\},\quad x_i\in E^n,\quad (i=1,\ 2,\ \cdots\cdots,\ n) \tag{7・3}$$

となることより明白である。また，分離型目的関数には，（７・２）式のような「和の型」の他に，

$$Z=F(x_1,\ x_2,\ \cdots\cdots,\ x_n)$$

$$=\prod_{i=1}^{n} f_i(x_i),\ x_i\in E^n \qquad (i=1,\ 2,\ \cdots\cdots,\ n) \tag{7・4}$$

のように，「積の型」になるものもある。この場合，「和の型」のように簡単には行かない。というのは，$f_i(x_i)$（$i=1,\ 2,\ \cdots\cdots,\ n$）に対する最大値は同様に，本論文の手法により容易に求めることができるけれども，各々の最大値の積が目的関数Ｚの最大値とは必ずしもならないからである。すなわち，一変数関数$f_1(x_1)$，$f_2(x_2)$，・・・・・，$f_n(x_n)$の最大値の中に負の値が存在する場合には，各々の積がＺの最大値とはならないことがある。

しかし，すべての目的関数の最大値が正であるならば，各々の最大値の積が目的関数Ｚの最大値になり得る。この意味において，「積の型」の分離型目的関数に関する最適化には，考慮の余地がある。実際の経営問題などでは，売り上げ高や企業利益のように正の範囲で最適値を求める問題が多いので「積の型」の場合でも，本論文の手法が充分に利用できるものと思われる。

8. 数値実験

第３章のアルゴリズムに従って，実際に数値実験を行なった。両方の計算において，探索区間は二等分割とした。結果は次に示す通り良好である。ここでアルゴリズムの停止条件は，相対誤差$\leqq 10^{-3}$であり，使用機種はＩＢＭシステム３６０／モデル１９５である。計算はすべて倍精度で行なった。

〔実験１〕

最大化；$f_1(x)=e^{-x}\cdot \sin x,\quad x\in[0,\ 16]$

リプシッツ定数；２

関数$f_1(x)$は極大値３個，極小値２個を持つ。収束計算における反復回数は，１７回であり詳しい結果は表１を参照されたい。

〔実験２〕

最大化；$f_2(x)=6x^5-15x^4-10x^3+30x^2+100,\quad x\in[-2,\ 2]$

リプシッツ定数；１２００

この場合，峰の個数は２個（極大値２個，極小値２個）である。反復回数は１６回で収束した。結果は表１の通りであるが，計算値が理論値と一致したのは，探索点がうまく理論値上に当った為と考えられる。

計算時間は実験１の場合が0.５７２７秒，実験２の場合が0.９６９４秒である。計算時間は探索区間の分割数およびリプシッツ定数，それに探索区間（変数の定義域）によって左右されるが，分割数に関しては経験より二等分割が良いようである。ここでいう計算時間とは，ＣＰＵタイムを指す。

本アルゴリズムの良さを確かめるために，ニュートン法との比較を試みた。ニュートン法は精度の良い方法として知られており，初期点を多く与えることにより多峰性関数によく利用される。$f_1(x)$，$f_2(x)$に適用したのであるが，表１で示した最大値が得られるまでに，前者は0.６６１０秒，後者は1.１０９２秒の計算時間を要した。我々のアルゴリズムは，ニュートン法と比較して計算時間が少

表１ 数値実験結果

	理論値	計算値	絶対誤差	最適解の存在範囲	最大誤差
f_1 (x^*)	0.32239694	0.32239693	1×10^{-8}	〔0.32239693, 0.32264108〕	7.572×10^{-4}
x^*	$\pi/4$	0.78527832	1.198×10^{-4}	〔0.78527832, 0.78540039〕	2.831×10^{-6}
f_2 ($\bar{x}$)	119	119	0	〔119, 119.07324〕	6.154×10^{-4}
$\bar{x}$	−1	−1	0	〔−1.0000610, −1〕	6.103×10^{-5}

（注） 関数f_1，f_2は，それぞれ実験１および２における目的関数であり，x^*，$\bar{x}$は最大値を与える最適変数である。絶対誤差は本論文の目的ではないが，この数値実験の場合，理論値が求まるので付記した。なお，一般の非線形最適化問題では，理論値が求まらないことが多いので絶対誤差は計算できない。「最大誤差」における数値は，「最適解の存在範囲」に対する相対誤差の最大値を示す。

表２ リプシッツ定数の及ぼす影響

項目＼ケース	Ⅰ	Ⅱ	Ⅲ	Ⅳ
リプシッツ定数	1	2	5	10
反復回数	16	17	18	19
CPUタイム（秒）	0.2806	0.5727	1.3140	2.8419

（注） ケースⅠはbest possibleなリプシッツ定数を与えた場合の実験結果を示す。

なくて済む。その上，真の解との誤差評価ができるという点がニュートン法や他の最適化手法と異なる大きな利点である。

次にリプシッツ定数が本アルゴリズムに対して与える影響を調べてみよう。実験１で使用した目的関数に対して，リプシッツ定数をbest possible な値，その２倍，５倍，および１０倍の場合についてアルゴリズムの収束状態を比較したのが表２である。表２から明らかなように，アルゴリズムが収束するための反復回数は，リプシッツ定数が少々過大であってもそれ程増えないことが分かる。またCPUタイムはbest possibleなリプシッツ定数を与えたときと比較して，他の３ケースとも指数的増加はみられない。よって本アルゴリズムでは，best possible なリプシッツ定数が得られなくとも，多少過大推定のリプシッツ定数を用いることにより最良解を得ることができる。図２は実験１および実験２における本アルゴリズムの収束状況を示したものである。

9. おわりに

以上より，リプシッツ定数による探索法は一変数多峰性関数に対して有効であることが判明した。

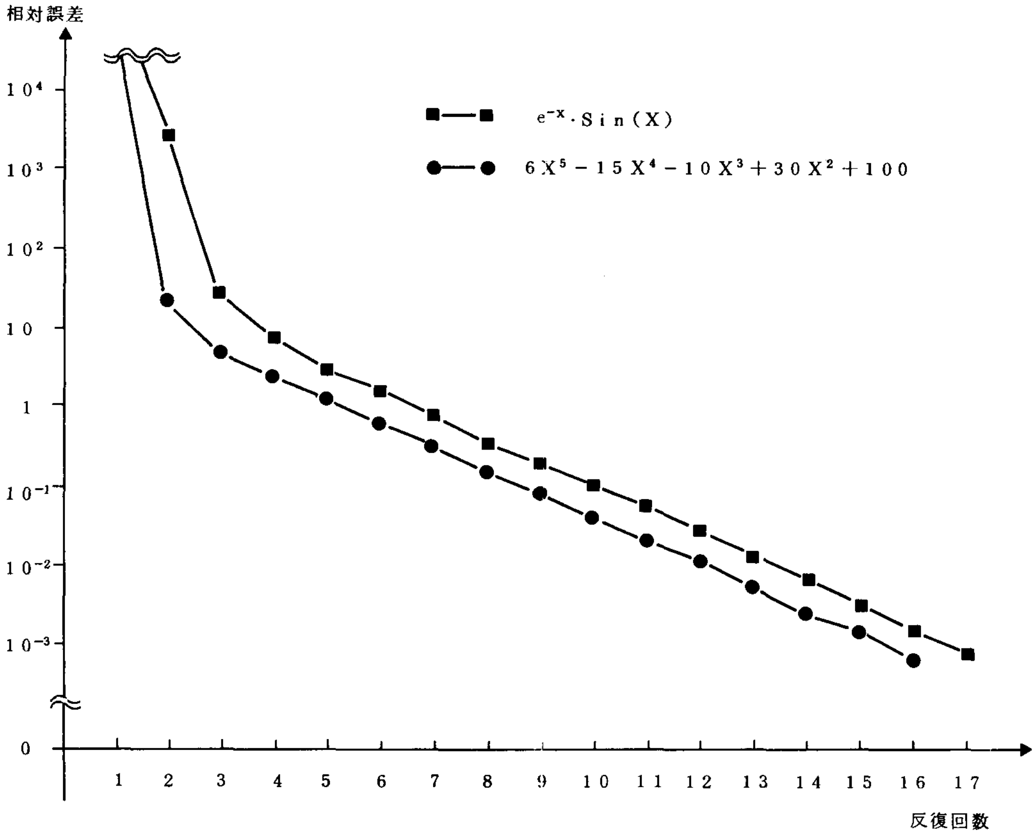

図2 アルゴリズムの収束状況

実験結果は表1で示したように，現実的には充分な精度の解が得られた。

現実の問題では，しばしば多変数多峰性関数の最適化問題に遭遇するが，多変数多峰性関数が分離型であれば，本アルゴリズムの適用が可能である。現在，関数近似論の世界では，すべての多変数連続関数を変数分離型で現わそうという試みがなされており[13]，それが完成すれば本アルゴリズムがそのまま適用できるものと思われる。

今後の研究課題として，多変数多峰性関数の最適値を求めるアルゴリズムへの拡張，リプシッツ定数のより良い決定方法などが残されている。

最後に，日ごろ御指導を賜わっている田口玄一教授，並びに有益な助言を与えて下さった矢頭攸介専任講師，馬渡鎮夫講師に深く感謝致します。

参考文献

〔1〕 Adby, P.R. and Dempester, M.A.H., Introduction to Optimization Methods, Chapman and Hall, 1974.

〔2〕 Branin, F.H., Jr., "Widely Convergent Method for Finding Multiple Solution of Simultaneous Nonlinear Equations," IBM J.Res.Dev. Sept. (1972), pp.504−522.

〔3〕 ―――――― and S.K.Hoo, "A Method for Finding Multiple Extrema of a Function of n Variables," Procceedings of the Conference on Numerical Methods for Nonlinear Optimization, University of Dundee, Scotland, June 28 − July 1, 1971, Numerical Methods for Nonlinear Optimization, Academic Press, London, August(1972), pp.231−237.

〔4〕 Brent, R.P., Algorithms for Minimization without Derivatives, Prentice−Hall, Inc., 1973.

〔5〕 Hadley, G.H., Nonlinear and Dynamic Programming, Addison − Wesley, 1964.

〔6〕 一松　信, "2変数函数を積の形で近似することについて，" 情報処理, 9, 1(1968), pp.14−17.

〔7〕 Kowalik, J. and Osborne, M.R., Methods for Unconstrained Optimization Problems, American Elsevier Publishing Company, Inc., 1968.

〔8〕 McMillan, C.Jr., Mathematical Programming : An Introduction to the Design and Application of Optimal Decision Machines, John Wiley & Sons, Inc., 1970.

〔9〕 Powell, M.J.D., "Recent Advances in Unconstrained Optimization," Mathematical Programming, 1, 1(1971), pp.26−57.

〔10〕 正道寺　勉, "多峰性関数に対する最適値探索の一解法，" 青山学院大学大学院理工学研究科経営工学専攻修士論文, 1975.

〔11〕 竹内　啓, 非線形計画法, 白日社, 1972.

〔12〕 津田孝夫, 佐藤美枝子, "多峰性多変数関数の最大・最小，" 情報処理, 16, 1(1975), pp.2−6.

脚　注

1） 参考文献〔1〕，〔7〕，〔9〕が詳しい。

2） 参考文献〔12〕によれば，「得られた解が大域的な最適解であることの保証を与えることはむつかしい。」とある。

3） 本論文中で用いる「最良解」とは，目的関数の目標値（真の最大値）にきわめて近い関数値を示すものであり，そのときの独立変数の値を指すものではない。

4） 目標値がきわめて大きい値，またはきわめて小さい値をとる目的関数を指す。これに関しては第5章を参照されたい。

5) 参考文献〔7〕を参照。

6) min{A, B}は，AまたはBの小さい方を選択するものとする。

7) 不等式(3・9)を満たさなかった区間は，最大値を見つける候補とはならないので，次の反復(iteration)では削除し，探索の効率を良くしている。したがって，探索が進むにつれ候補となる区間は少なくなる。ここで注意すべきは，3・2で示すように，区間幅hはiterationごとに小さくなることである。本アルゴリズムの区間縮小比率は1／nであるから，第kステップにおける区間幅$h^{(k)}$は$(b-a)/n^{(k-1)}$ $(k \geqq 2)$となる。

8) (j)はiterationの回数を示す。

9) i^*はステップ5において，YSTAR≦YMAXを満足するようなiを示す。

10) すなわち，目的関数を$g(x)=f(x)\pm$ const.として$g(x)$を最適化すればよい。

11) 参考文献〔10〕pp.81−86では，フーリエ級数展開にチェビシェフ多項式を導入することによりリプシッツ定数を求めている。それによれば，リプシッツ定数は目的関数の係数の絶対値の総和掛ける次数の2乗で与えられることが示されている。目的関数が微分不可能な場合には，リプシッツ定数を決定するための有効な手法であると思われる。

12) 詳しくは参考文献〔5〕，pp.136−141を参照。

13) 例えば，参考文献〔6〕を参照。