

多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 GLOBAL OPTIMA FOR MULTI-VARIABLE MULTI-MODAL FUNCTIONS

TSUTOMU SHOHDOHJI
Aoyama Gakuin University

(Received April 19, 1977; Revised July 18, 1977)

Abstract. Optimization problems for non-linear systems often involve those cases which preclude the assumption of uni-modality of the objective functions. The methodology has been so far developed with regards to optimizing of multi-variable multi-modal objective functions. It has, however, failed in guaranteeing the global maxima or minima of solutions.

This study is an extension of the search method previously developed by the author for single-variable multi-modal functions to multi-variable multi-modal objective functions. The algorithm presented in this paper searches the global optima under the Lipschitz condition. It provides the approximation of the global optima and the upper bound of the global maxima. Through numerical tests, it is shown that the algorithm efficiently produces satisfactory results.

1. ま え が き

非線形最適化手法については今日まで多くの研究がなされているが、大域解であることを保証する決定論的な方法はまだ報告されていないようである〔6〕。古くから非線形最適化問題に対して適用できると考えられていた方法には極値探索法と呼ばれるものがあるが、これらのほとんどが探索領域内で目的関数の単峰性を仮定しているものであり、多峰性関数には適用が困難なものである。その中で多峰性関数に適用できる方法としてはランダム探索法〔4〕、修正ランダム法〔11〕、完全探索法〔5〕、それに多峰性関数の探索領域を分割して各領域内では単峰性を仮定し、各領域内で単峰の場合の手法を用いるタブレーション法〔8〕などが挙げられる。これらの方法では得られた解が大域解であることの保証が与えられなればかりか、探索効率の点からも改善すべき点があるように思われる。特に峰の数や変数の数が増せば増すほど効率が悪くなると予想される。

これに対し、最近の研究では極値線をネットワーク状に探索する手法〔7〕、〔9〕、峰を分離する方法〔12〕、関数のあてはめによる方法〔10〕、微分方程式の軌跡を追跡する方法〔1〕、〔2〕、〔3〕、

ランダムサンプリング法に人間の判断を組み入れた方法〔14〕などがある。これらの方法も得られた解が大域解であるという保証を与えることは困難であるように思われる。

本研究は〔13〕で述べた方法を多変数多峰性関数に適用できるよう拡張したものであり、連続的に微分可能な多変数多峰性の目的関数を有界閉領域内で最大化したい場合に適用できる有効な探索アルゴリズムおよび計算結果を示す。なお本研究での最適解（目標値を指す）とは、工学上での最適解である。すなわち、工学上は真の大域的 maximum とほとんど差のない解（本論文ではその保証を与えている）が得られれば良いのである。

2. アルゴリズムの基本的な考え方

2・1 記号の説明

アルゴリズムの基本的な考え方について述べる前に本研究で考察する問題を設定し、そこで使用する記号の定義をしておこう。

我々は目的関数を最大化することを考えるが、最小化したい場合には目的関数の符号を変換することにより容易に最大化問題となるので、本研究では最大化問題のみについて考える。

目的関数の一般的な形を $Y = f(X)$, $X = (x_1, x_2, \dots, x_n)$ とし、 f は連続的に微分可能なすなわち $f(X) \in C^1(Q)$ であるか又は $g(X) \in C^1(Q)$ によって $f(X) = |g(X)|$ と表わされる関数である。ここで探索領域 Q は n 次元区間: $[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$ である。このとき目的関数が大域的 maximum となる点を $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ とすれば、点 X^* における目的関数の値（真の maximum） Y^* は、

$$Y^* = f(X^*) = \max_{X \in Q} f(X)$$

となる。

探索領域 Q は探索アルゴリズムの反復ごとに小さな領域に分割され、我々は分割された小領域のことを分割小領域と呼ぶ。アルゴリズムの第 k 反復における各々の分割小領域は点 $P^k = (P_1^k, P_2^k, \dots, P_n^k)$ を中心とする n 次元矩形 $A(P^k)$ であり、次のような集合である。

$$A(P^k) = \{ (q_1, q_2, \dots, q_n) \in Q : |q_i - P_i^k| \leq \frac{\ell_i}{2^{k-1}}, i = 1, 2, \dots, n \}$$

ここで、 $\ell_i = b_i - a_i$ ($i = 1, 2, \dots, n$) である。

2・2 探索方法の着眼点

我々は与えられた目的関数がどんな形状をしているかが全く分からなくても、目的関数を最大にしたという必要性にしばしば直面する。この場合、適用できる方法の1つとして古くからよく知られている完全探索法（格子点探索法）を挙げることができる。この方法は各変数ごとに定義域を分割し、そのメッシュを小さくすることにより精度を高めようとする方法である。しかしながら、これには大きな欠点がある。1つは、いくらメッシュを細かくしても、得られた解が大域解であるという保証を与えることができないことであり、もう1つは、解を計算する際に大型コンピュータを利用する

としても記憶容量および計算時間は有限であるから、目的関数の変数が多いものについては適用が困難である。そこでこれらの欠点を克服するために、Lipschitz条件を活用する。

簡単のため2変数の場合のLipschitz条件について述べる。

定義 1. 関数 f とその定義域 Q にのみ依存する定数 $L > 0$ が存在して、任意の2点 (x_1, y_1) , $(x_2, y_2) \in Q \subset R^2$ が与えられたとき、つねに

$$(2 \cdot 1) \quad |f(x_1, y_1) - f(x_2, y_2)| \leq L \cdot \|(x_1, y_1) - (x_2, y_2)\|_2$$

を満足するとき、関数 f はLipschitz条件を満たすという。そしてこの L をLipschitz定数と呼ぶ。ここで $\|\cdot\|_2$ は2次元Euclidノルムを示す(2・3を参照)。

不等式(2・1)は変数の数が増えても同様に拡張できる。定義1を利用すれば、大域性を保証(2・4を参照)しながら近似解を得ることができる。では具体的にどのようにして探索を行えばよいかを次に示す。

探索領域 Q を図1で示した領域 (Q は正方領域である必要はなく、 a_i, b_i は負でもよい)とする。

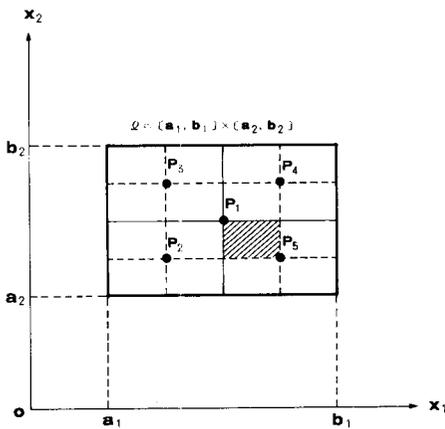


図1. 探索点の生成(2変数の場合)

点 P_1 は各辺の中点を結んだ交点、すなわち探索領域 Q の中心点であり、点 P_1 において収束条件(2・4で述べる)を満足していなければ分割を行なう。このとき、点 P_1 に対して4個の探索領域が生成され、それぞれの中心点(P_2, P_3, P_4, P_5)が新しい探索点となる。これらの探索点に対して、探索点を含む領域をさらに分割して探索する必要があるかを調べる。以下同様の議論をすすめて、収束条件を満足すればアルゴリズムは停止する。3変数の場合には、図2のように探索点1個に対して 2^3 個の探索点が生成される。探索方法は n 変数の場合でも、2変数の場合と同じやり方でよい。

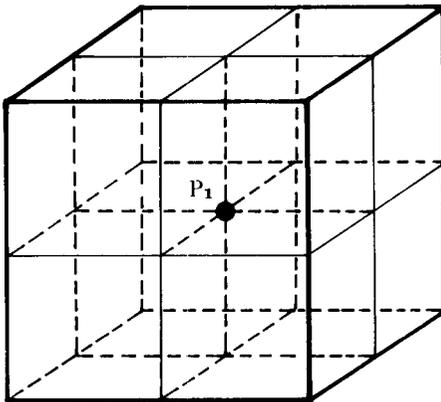


図2. 探索点の生成(3変数の場合)

本研究では、Lipschitz条件を利用して探索する必要のない領域は各反復ごとに捨てており、これがアルゴリズムの効率を良くしている。

2.3 分割小領域内の最大半径

2.2では探索領域 \mathcal{Q} をある判断基準の下で探索領域の分割を行なうことを示した。この節では、その判断基準について述べる準備として、分割小領域内の最大半径について述べる。図3で示したよ

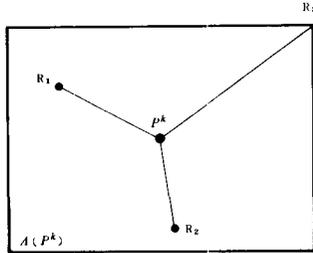


図3. $A(P^k)$ 内の最大半径

うに、点 P^k を中心とする分割小領域 $A(P^k)$ (たとえば図1で斜線を付けた領域を考えればよい)内の任意の点 R と中心点 P^k との距離を $\rho(P^k, R)$ とすれば、 $\rho(P^k, R)$ の最大値を与える点は、図3より明らかに、対角線上の点 R_3 である。このことは、すべての分割小領域についていえる。ここで $A(P^k)$ 内の点 P^k を中心とする最大半径を求めておこう。

本アルゴリズムの第 j 反復(すなわち第 j 回目の分割)における分割小領域 $A(P^j)$ 内の最大半径 M_j は、

$$(2.2) \quad M_j = \left\{ \left(\frac{\ell_1}{2j} \right)^2 + \left(\frac{\ell_2}{2j} \right)^2 \right\}^{\frac{1}{2}}$$

で表わせる。ただし、 $\ell_1 = b_1 - a_1$ 、 $\ell_2 = b_2 - a_2$ である。これは2変数の場合であるが、 n 変数の場合は次のようになる。

$$(2.2)' \quad M_j = \left\{ \sum_{i=1}^n \left(\frac{\ell_i}{2j} \right)^2 \right\}^{\frac{1}{2}} \quad (\ell_i = b_i - a_i, \quad i = 1, 2, \dots, n).$$

2.4 探索領域分割のための判断基準(大域性の保証)

本アルゴリズムは、大域解を含む可能性のある領域についてのみ探索を行なう。ここで重要なことは、ある分割領域に対して、さらにその領域を探索する必要があるかないかの判定を正しく行なうことである。この判断基準となっているのが次の定理1である。

定理1. $f_i^{(j)}$ を第 j 反復において生成された探索点 P_i^j (その個数を m_j とする)における関数値、 $f_{\max}^{(j)}$ を第 j 反復までの最大値(3.2のステップ3を参照)、 L をLipschitz定数とする。第 j 反復における各分割小領域 $A(P_i^j)$ 内の $f_i^{(j)}$ に対して、

$$(2.3) \quad f_{\max}^{(j)} > f_i^{(j)} + L \cdot M_j \quad (i = 1, 2, \dots, m_j)$$

となる $f_i^{(j)}$ があれば、 P_i^j を含む分割小領域 $A(P_i^j)$ 内には $f_{\max}^{(j)}$ よりも大きな値が存在しない。

(証明) 任意の点 $R \in A(P_i^j)$ をとれば、定義1より、

$$(2.4) \quad f(R) - f(P_i^j) \leq L \cdot M_j$$

が成立する。これを变形すれば、

$$(2.5) \quad f(R) \leq f(P_i^j) + L \cdot M_j$$

となる。不等式(2.3)を満足する $f_i^{(j)}$ が存在すれば、

$$(2.6) \quad f(P_i^j) + L \cdot M_j < f_{\max}^{(j)}$$

であるから、不等式(2.5)と(2.6)より、

$$(2.7) \quad f(R) < f_{\max}^{(j)}$$

である。Q. E. D.

この定理1は Lipschitz 定数 L の利用により、捨てる領域を決定できることを意味している。すなわち、Branch and Bound 法の陰的列挙に相当するものである。本アルゴリズムの良さは完全探索法などと異なり、探索領域全域を実際に調べなくても陰的に調べている点（つまり、捨てる分割小領域の中にはそれまでの暫定解よりも大きな値を与えないことが証明される点）にある。

3. アルゴリズム

3.1 Lipschitz 定数の求め方

本アルゴリズムは、定義1で述べた Lipschitz 条件を活用するので、Lipschitz 条件が成り立つような Lipschitz 定数を求める必要がある。この Lipschitz 定数の最小値が得られれば良いのであるが、これは一般に困難である。本研究の目的は与えられた多変数多峰性の目的関数を最大化することにあるので、最小の Lipschitz 定数を求める必要はない¹⁾。したがって、ここでは1つの Lipschitz 定数を求めるための1つの方法を述べることにする。

f を連続的に微分可能な目的関数、 \mathcal{Q} を探索領域である n 次元区間としたとき、

$$(3.1) \quad |f(X) - f(Y)| \leq L \cdot \|X - Y\| \text{ for all } X, Y \in \mathcal{Q}$$

となる1つの定数 L を求めたい。そのために次の考察をする。

$X, Y \in R^n$ ($X \neq Y$) を任意にとりて固定し、 $Z = (z_1, z_2, \dots, z_n)$ とおけば、平均値の定理より、

$$(3.2) \quad f(X+Z) = f(X) + \sum_{k=1}^n z_k \cdot \frac{\partial f}{\partial x_k}(x_1 + \theta z_1, x_2 + \theta z_2, \dots, x_n + \theta z_n) \quad (0 < \theta < 1)$$

を得る。したがって、(3.2)式を変形して両辺を $\|Z\|$ で割り、Cauchy-Schwarz の不等式を適用すれば、

$$(3.3) \quad \frac{|f(X+Z) - f(X)|}{\|Z\|} \leq \left(\sum_{k=1}^n \left| \frac{\partial f}{\partial x_k}(x_1 + \theta z_1, x_2 + \theta z_2, \dots, x_n + \theta z_n) \right|^2 \right)^{\frac{1}{2}} \quad (0 < \theta < 1)$$

$$(3.4) \quad \leq \max_{U \in \mathcal{Q}} \left(\sum_{k=1}^n \left| \frac{\partial f}{\partial x_k}(U) \right|^2 \right)^{\frac{1}{2}}$$

を得る。よって、

$$(3.5) \quad \max_{U \in \mathcal{Q}} \left(\sum_{k=1}^n \left| \frac{\partial f}{\partial x_k}(U) \right|^2 \right)^{\frac{1}{2}} \leq L$$

1) 文献[13]では、Lipschitz 定数の与える影響について考察がなされており、これによれば最適（すなわち最小）な定数が得られなくても、アルゴリズムにはそれ程大きな影響を及ぼさないことが示されている。このことは多変数の場合についてもいえる。

となる定数 L は 1 つの Lipschitz 定数である (本アルゴリズムでは不等式 (3.5) の等号が成り立つ数値を採用している)。

3.2 探索アルゴリズム

2 で述べた考えに基づく探索の具体的計算手順は次のとおりである。

ステップ 1: (初期値設定) Lipschitz 定数 L , 変数の数, Stopping Rule 定数 ε , 各変数の定義域, 最初の探索点 $x_1^{(1)}$ を入力し, $i = 1, j = 1, m_i = 1$ とする。

ステップ 2: 探索点 $x_i^{(j)}$ における関数値 $f_i^{(j)} = f(x_i^{(j)})$ ($i = 1, 2, \dots, m_j$) を求める。

ステップ 3: ステップ 2 で求めた $f_i^{(j)}$ の中での最大値 $f_{\max}^{(j)}$ を見つける;

$$f_{\max}^{(j)} = \begin{cases} f_1^{(1)} & (j = 1) \\ \max_{1 \leq i \leq m_j} [\max_{1 \leq i \leq m_j} f_i^{(j)}, f_{\max}^{(j-1)}] & (j \geq 2) \end{cases}$$

ステップ 4: 分割小領域 $A(x_i^{(j)})$ 内の最大半径 M_j を求める。

ステップ 5: (収束の判定) 収束判定のための条件式;

$$\left| \frac{L \cdot M_j}{f_{\max}^{(j)}} \right| \leq \varepsilon$$

を満足すれば, ステップ 7 へ, 満足しない場合にはステップ 6 へ進む。

ステップ 6: (探索領域の改善)

$$f_{\max}^{(j)} \leq f_i^{(j)} + L \cdot M_j$$

を満足するような点 $x_i^{(j)}$ を含む領域を全て見つけ, $j \leftarrow i + 1$ と置き換え, 探索点 $x_i^{(j)}$ ($i = 1, 2, \dots, m_j$) を決定し, ステップ 2 へ戻る。

ステップ 7: (計算終了) 真の最大値 Y^* は,

$$f_{\max}^{(j)} \leq Y^* \leq f_{\max}^{(j)} + L \cdot M_j$$

なる範囲にあり, $f_{\max}^{(j)}$ (およびそのときの $x_i^{(j)}$), M_j を出力する。

3.3 誤差評価

本アルゴリズムの特徴の 1 つは誤差評価ができることにある。探索アルゴリズムが第 k 反復で終了したものとすれば, 3.2 のステップ 7 より真の最大値 Y^* は,

$$(3.6) \quad f_{\max}^{(k)} \leq Y^* \leq f_{\max}^{(k)} + L \cdot M_k$$

の範囲にある。以下 $f_{\max}^{(k)} > 0$ として議論を進めることにする。不等式 (3.6) の辺々から $f_{\max}^{(k)}$ を引けば,

$$(3.7) \quad 0 \leq Y^* - f_{\max}^{(k)} \leq L \cdot M_k$$

となり, 相対誤差は,

$$(3.8) \quad \left| \frac{Y^* - f_{\max}^{(k)}}{Y^*} \right| \leq \left| \frac{L \cdot M_k}{Y^*} \right|$$

となる。不等式(3.8)では Y^* が未知であるため両辺とも計算ができない。そこで $f_{\max}^{(k)} \leq Y^*$ を利用すれば、

$$(3.9) \quad \left| \frac{L \cdot M_k}{Y^*} \right| \leq \left| \frac{L \cdot M_k}{f_{\max}^{(k)}} \right|$$

が成立するので、右辺は相対誤差の上界である。よって、本アルゴリズムでは相対誤差の評価ができる。一方、 $f_{\max}^{(k)} < 0$ の場合には、不等式(3.6)の右辺を不等式(3.8)に適用して、

$$(3.10) \quad \left| \frac{L \cdot M_k}{Y^*} \right| \leq \left| \frac{L \cdot M_k}{f_{\max}^{(k)} + L \cdot M_k} \right|$$

を得る。この右辺が $f_{\max}^{(k)} < 0$ の場合の相対誤差の上界となる。また $f_{\max}^{(k)} = 0$ の場合には、相対誤差評価ができないので目的関数にある正の定数を加え不等式(3.9)による相対誤差評価で収束の判定を行なうか、3.2のステップ5を $|f_{\max}^{(j)} - f_{\max}^{(j-1)}| \leq \epsilon$ に置き換えることによりアルゴリズムおよび相対誤差評価の議論は一般性を失わない。

4. 数 値 実 験

次に示す5つのテスト関数(すべて最大値を見つけることを目的としている)について本アルゴリズムを適用した。テスト関数(1)~(4)は参考文献[10]からの引用であり、表1および表2では文献[10]での計算結果と本アルゴリズムによる計算結果との比較が示してある。計算に使用したコンピュータはIBM370/M168である。テスト関数(5)に関しては、アルゴリズムの停止定数を $\epsilon=10^{-2}$ (すなわち、相対誤差の最大値が1%以内であることを示す)としている。

<テスト関数>:

- (1) $f_1(x, y) = 25 \text{EXP} \{ -20(x-0.3)^2 - 18(y-0.7)^2 \} + 23 \text{EXP} \{ -17(x-0.65)^2 - 19(y-0.25)^2 \}$
- (2) $f_2(x, y) = 18 \text{EXP} \{ -15(x-0.5)^2 - 20(y-0.7)^2 \} + 19 \text{EXP} \{ -22(x-0.27)^2 - 20(y-0.25)^2 \} + 17 \text{EXP} \{ -20(x-0.75)^2 - 16(y-0.3)^2 \}$
- (3) $f_3(x, y) = 15 \text{EXP} \{ -20(x-0.3)^2 - 22(y-0.3)^2 \} + 17 \text{EXP} \{ -19(x-0.75)^2 - 15(y-0.25)^2 \} + 14 \text{EXP} \{ -23(x-0.25)^2 - 18(y-0.75)^2 \} + 16 \text{EXP} \{ -20(x-0.7)^2 - 20(y-0.8)^2 \}$
- (4) $f_4(x, y) = \{ 1 - 100(x-0.15)(x-0.35)(x-0.5)(x-0.95) \} \times \{ 1 - 100(y-0.1)(y-0.3)(y-0.6)(y-0.95) \}$
- (5) $f_5(x, y, z) = | \sin x + \cos y + \sin x \cdot \cos z | + 100$

<探索領域>:

(1)~(4)については $[0, 1] \times [0, 1]$, (5)は $[-3.5, 3.5] \times [-3.5, 3.5] \times [-3.5, 3.5]$ である。

<Lipschitz constant>; (1) 52.93, (2) 68.31, (3) 138.2, (4) 9.5, (5) 2.45

表1の相対誤差をみれば分かる通り、本アルゴリズムによる精度の方が優れており、表2で示したように探索点の数はそれほど多くない²⁾。表3ではテスト関数(5)の計算結果を示してあるが、探索アルゴリズムの反復回数は5回、計算時間(CPU Time)は0.38秒である。この場合、真の最大値 Y^* は分かっているが、本アルゴリズムでは下界と上界が求まるので Y^* は $102.96487 \leq Y^* \leq 103.62148$ の範囲に必ず入ることが保証され、 Y^* が未知の場合には威力を発揮するものと思われる。テスト関数(5)に完全探索法を適用し、表3で示した計算結果と同じ精度の解を得るには29,791個の探索点を必要とした。本アルゴリズムでは1,161個の探索点であったから完全探索法の約3.88%で済むことが分かる。

表1. テスト関数(1)~(4)に関する計算結果の比較表 (注)

テスト関数		(1)	(2)	(3)	(4)
項目					
計算結果(I)	x	0.30078	0.27734	0.74219	0.80469
	y	0.69922	0.25391	0.25781	0.82031
	f	25.06195	19.32002	17.29747	4.80012
計算結果(II)	x	0.2980	0.2755	0.7485	0.8065
	y	0.6979	0.2500	0.2515	0.8229
	f	25.052	19.315	17.291	4.789
真の解	x	0.301	0.2750	0.742	0.807
	y	0.699	0.2540	0.253	0.822
	f	25.062	19.321	17.303	4.800
(I)の相対誤差	(%)	0.00020	0.00507	0.03196	0.0
(II)の相対誤差	(%)	0.03990	0.03105	0.06935	0.22917
CPU Time (I)	(秒)	0.25	0.16	0.19	0.24

(注) (I)は本アルゴリズムによる計算結果を示し、(II)および真の解は久保田・北森[10]による。

テスト関数(4)では、本アルゴリズムで得られた結果の方が真の解で示されている値よりも大きな関数値が得られたけれども、これは計算機内での有限桁計算の関係だと思われる。CPU Timeは本アルゴリズムによる計算時間である。

2) この計算結果は久保田・北森の方法と比較したものであるが、表2の(注)で示したように、比較の方法は久保田・北森によるアルゴリズムの結果と同等もしくはそれ以上の精度の解が得られたところで我々のアルゴリズムを打ち切った場合、どれだけ探索点を必要としたかを考えている。

表 2. テスト関数(1)~(4)の探索点比較表^(注)

項目 \ テスト関数	(1)	(2)	(3)	(4)
我々の探索法	85	109	65	81
久保田・北森氏の方法*	85	97	89	74
ランダム探索法*	590	906	764	304

(注) *印は文献[10]から引用した。本アルゴリズムによる探索点は、表1の計算結果(II)で示された値よりも良い値を得るまでに必要とした探索点の個数を示す。

表 3. テスト関数(5)の計算結果

	x	y	z	$f(x, y, z)$	相対誤差(%)
理論値	-1.57080,	-3.14159,	0	103	—
計算結果	-1.53125,	-3.28125,	-0.21875	102.96487	0.031067
	-1.53125,	-3.28125,	0.21875	102.96487	
	-1.53125,	3.28125,	-0.21875	102.96487	
	-1.53125,	3.28125,	0.21875	102.96487	

5. あとがき

本研究は一変数多峰性関数の大域的な最大値探索に関する手法を多変数多峰性関数に適用できるよう拡張したものであるが、数値実験結果を見れば分かるように、実用上十分な精度の解が得られた。本アルゴリズムの大きな特徴は次の3つである。

- ① 得られた解が大域的な解(最大値)の高精度な近似値であることの保証を与えている。
- ② 大域的な最大値(一般に未知)の下界および上界を希望の精度で定めることができる。
- ③ 相対誤差の評価ができる。

今後の研究課題としては、目的関数の種類により探索領域の分割方法を考慮する必要があること、および変数が大幅に増加した場合に起こる困難(記憶容量、計算時間の増加など)に対する考慮、Lipschitz定数の各反復ごとにおける改善などが残されている。また昭和52年3月18日の日本OR学会春季研究発表会にて、北海道大学関口恭毅助教授より本研究とBranch and Bound法との関連について有益なるコメントを頂きましたことに感謝し、今後この方面の研究も進めたいと思っている。

最後に、日頃ご指導を賜わっている本学田口玄一教授ならびに有益な助言を与えて下さった論文審査員の方々に深く感謝致します。

参 考 文 献

- [1] Branin, F. H. Jr. and Hoo, S. K., "A Method for Finding Multiple Extrema of a Function of N Variables," in *Numerical Methods of Nonlinear Optimization*, Academic Press, London, pp. 231-237, Aug. (1972).
- [2] Branin, F. H. Jr., "Widely Convergent Methods for Finding Multiple Solutions of Simultaneous Nonlinear Equations.," *IBM J. Res. Develop.*, pp. 504-522, Sept. (1972).
- [3] Brent, R. P., "On the Davidenko-Branin Method for Solving Simultaneous Nonlinear Equations.," *IBM J. Res. Develop.*, pp. 434-436 July (1972).
- [4] Brooks, S. H., "A Discussion of Random Methods for Seeking Maxima," *Operations Research*, vol. 6, No. 2, pp. 244-251 (1958).
- [5] Dixon, L. C. W. / 松原正一訳, 非線形最適化計算法, 培風館, p. 31, 1974.
- [6] ————— and Szegö, G. P. ed., *Towards Global Optimisation*, North-Holland, 1975.
- [7] 榎本肇, 片山卓也, 川本栄二, "リンク手法による多変数関数の極値探索," 情報処理, vol. 17, No. 9, pp. 835-842 (1976).
- [8] Hill, J. D., "A Search Technique for Multimodal Surfaces," *IEEE Trans. System Sci. and Cybernetics* 5, pp. 2-8 (1969).
- [9] 久保洋, "多峰性最適化問題の一解法," 計測自動制御学会論文集, vol. 8, No. 6, pp. 56-61 (1972).
- [10] 久保田健二, 北森俊行, "多次元多峰性未知関数の最大値探索法," 計測自動制御学会論文集, vol. 10, No. 6, pp. 67-74 (1974).
- [11] McMurtry, G. J. and Fu, K. S., "A Variable Structure Automaton Used as a Multimodal Searching Technique," *IEEE AC-11*, No. 3, pp. 379-387 (1966).
- [12] 沖野教郎, 佐々木泰生, "モンテカルロ積分を利用した多峰性関数の各峰分離——多峰性極値探索問題の一解法——," 計測自動制御学会論文集, vol. 8, No. 6, pp. 48-55. (1972).
- [13] 正道寺勉, "一変数多峰性関数に対する最適値探索法の研究," 日本オペレーションズ・リサーチ学会論文集 (JORSJ), vol. 19, No. 4, pp. 295-307 (1976).
- [14] 津田孝夫, 佐藤美枝子, 大久保英嗣, 吉田秀逸, "多峰性多変数関数の最大・最小と図形会話型処理," 情報処理, vol. 17, No. 9, pp. 812-819 (1976).

著者連絡先: 青山学院大学理工学部

〒157 東京都世田谷区千歳台6の16の1 TEL: (03) 300-2121
Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.