単峰領域の概念に基づく一変数多峰性関数の複数極大点および最大点探索手法

金光 秀雄[†] 新保 勝[††]

有限個の極大点を区間内部に有する一変数多峰性関数の最大化問題において，極大点の近傍での性質を示す．この性質から，各極大点での単峰領域と単峰領域半径を新たに定義する．単峰領域の定義から，3点が単峰領域上にある場合に両端の2点が極大点を含む十分条件を導く．次に，この関係を利用して，一定間隔で与えられた点列と各点での関数値から，各極大点を囲む隣接3点を検出する手法（複数極大点検出手法）を提案する．また，この手法がある極大点を検出するための条件を単峰領域上に隣接する4点の関係として与え，この手法が最大点を囲む隣接3点を検出する条件とすべての極大点を囲む隣接3点を検出する条件を点の間隔と単峰領域半径を用いて導く．

複数極大点検出手法で得られた極大点を囲む隣接3点の集合から，次の2手法：(1) 各隣接3点に対して局所最適化手法を適用して，複数極大点を与えられた要求精度で求める手法，(2) これまで得られた最大値を超える値が得られる可能性のない3点を取り除き，最大値を与える可能性のある3点だけに局所最適化手法を適用することにより，与えられた要求精度で効果的に最大点を求める手法を提案する．数値実験により本手法と他の手法と比較し，提案した2手法が複数極大点あるいは最大点を効果的に見いだすことを示す．複数極大点を探索する手法は非常に単純であるため，容易に実現可能である．

Two Methods for Finding Local and Global Maxima of a Multimodal Function of One Variable Based on the Concept of a Unimodal Region

HIDEO KANEMITSU[†] and MASARU SHIMBO[††]

We consider a maximization problem of a univariate continuous function on an interval that has finite number of local maxima within the interval. We discuss properties in the neighbourhood of a local maximum. Then, we define an unimodal region and its radius of a local maximum. For three points exist in some unimodal region, we derive a sufficient condition such that a local maximum exists between these two end points. Using the condition, we propose an algorithm for detecting the set of elements which consist of three neighbouring points of which two end points include each local maximum. We show the relationship among four functional values on a unimodal region for detecting the subinterval that includes a local maximum. We derive two conditions for including global maxima and all local maxima. Using the algorithm and a local optimizer, we propose an algorithm for finding multiple local maxima and an algorithm for finding global maxima by effectively removing three such points that cannot find a global maximum. The results of numerical experiments show that our two methods are effective for finding global maxima or other local maxima.

1. はじめに

多峰性関数の最大点を含む複数の極大点を求める手法（大域的最適化手法）については，確率論的な手法と決定論的な手法に大きく分けられる．

確率論的な手法の中で，SA 法[1),8)]や GA 法[6)]は新しい枠組みの手法として最近研究がさかんであるが，これらの手法は理論的な背景が十分でないため，最大点を見いだす条件が得られておらず，効率も一般に良くない．ベイズ法による手法[10),15),16)]も確率論的な手法として効率も良いが，アルゴリズムが複雑なため，構成が困難で関数評価以外の処理に時間を要するという問題点がある．

決定論的な手法では，Lipschitz 条件を用いた手法

† 北海道教育大学函館校情報科学教室
Computer and Information Sciences Laboratory, Hakodate Campus, Hokkaido University of Education
†† 北海道大学工学部情報工学科
Department of Information Engineering, Faculty of Engineering, Hokkaido University

や区間演算法を用いた手法がある．Lipschitz 条件を用いた手法[3),4),9),11),12)]は，アルゴリズムがやや複雑になることと，Lipschitz 定数の与え方が難しく，その与え方によっては効率が低下してしまう[13)]．区間演算法を用いた手法[5),7)]は解を含む区間を高精度に求めることができるという特徴があるが，区間演算を行う環境の実装が一般には困難であり，効率もあまり良くない．

一方，単峰性関数の最大点を求めるための局所最適化手法はアルゴリズムとして完成され，十分な理論的背景のもとに強い収束性が得られており，効率的な手法が数多く存在する．両者の手法の効率性や構成の違いとなる大きな原因として，単峰性関数の最大化問題では，単峰性関数のサブクラスとしての凹（凸）関数等の性質が十分研究されており，これらの性質を利用して効率の良いアルゴリズムが構成されているが，多峰性関数の最大化問題では，多峰性関数の定義や性質が知られていない[14)]ため，その性質を利用した効率的なアルゴリズムが構成できないことがあげられる．

本論では，有限個の極大点を有する一変数多峰性関数の極大点の近傍での性質や，単峰領域と単峰領域半径の定義を新たに示し，これらの性質や定義から導かれる関数値間の関係を利用して多峰性関数の複数極大点を検出し，各極大点の単峰領域上で効率的でかつライブラリ等で容易に利用可能な局所最適化手法を効果的に組み合わせることにより，全体として容易に実装可能でかつ効率的な決定論的アルゴリズムの構成を目的としている．

2 章で，準備として問題の定義と記法を与える．3 章では極大点の近傍で成立する性質を導き，この性質から，各極大点での単峰領域および単峰領域半径を定義し，この単峰領域上の点が極大点を含むための条件を関数値間の関係として与える．4 章では，3 章で得られた単峰領域上の関数値間の関係を利用して，各極大点を含む 3 点を検出するためのアルゴリズムを構成し，本アルゴリズムが最大点を含む 3 点とすべての極大点を含む 3 点を見いだす条件を導く．さらに 5 章で，4 章のアルゴリズムと局所最適化手法を用いて，複数極大点を要求精度で求めるアルゴリズムと，最大点を要求精度で求めるアルゴリズムを示す．6 章では，いくつかのテスト関数による数値実験から本手法と他の手法を比較し，考察する．最後に 7 章でまとめと今後の課題を述べる．

2. 準　　備

2.1 問　　題

本論では，次式

$$\left\{\begin{array}{l} f_o(x) \to max, \quad x \in D = [a,b] \subseteq R \\ \text{where}\,,\ f_o\ :\ D \to R \end{array}\right. \qquad (\mathrm{P}')$$

で表される一変数多峰性関数の大域的最大化問題 (P′) を扱う．ここで，問題 (P′) の目的関数 $f_o(x)$ は連続であることを仮定する．また，問題 (P′) の関数の定義域を R 全体として考えるために，拡張された目的関数 $f(x)$

$$f(x) = \left\{\begin{array}{ll} (x-a)+f_o(a), & x < a \\ f_o(x), & a \le x \le b \\ -(x-b)+f_o(b), & x > b \end{array}\right. \quad (1)$$

を考え，この関数を R 上で最大化する問題を本論で扱い，これを問題 (P) とする．

(注意 1) 関数 f は $x < a$ で単調増加，$x > b$ で単調減少する連続な関数なので，極大点は問題 (P′) と同一で D 上だけに存在する．

問題 (P) の極大点は，ある $\delta > 0$ が存在して

$$f(\overset{*}{x}) \ge f(x), \quad \forall x \in (\,\overset{*}{x} - \delta, \overset{*}{x} + \delta\,) \quad (2)$$

を満足する点として定義され，極大点の集合を $\overset{*}{X}$ とする．さらに，問題 (P) の極大点に関して以下の仮定をする．

仮定 1 相異なる 2 点の極大点がある一定値以上離れている．すなわち，ある $\alpha > 0$ が存在して

$$\min_{\overset{*}{x} \neq \overset{*}{y} \in \overset{*}{X}} \{|\,\overset{*}{x} - \overset{*}{y}\,|\} = \alpha \quad (3)$$

であるとする．

仮定 1 は，問題 (P) の極大点が有限個存在することと同等であるから，これらの極大点を $\overset{*}{x}{}^j$ ($j = 1, 2, \ldots, m$) と記すことにする．

(注意 2) 仮定 1 を満足する極大点 $\overset{*}{x}$ は，ある $\delta > 0$ が存在して

$$f(\overset{*}{x}) > f(x), \ \forall x \neq \overset{*}{x} \in (\,\overset{*}{x} - \delta, \overset{*}{x} + \delta\,)$$

で表される狭義の極大点の定義を満足する．

また，問題 (P) の極大点での関数値の集合を $\overset{*}{F}$ で表し，最大点を $\overset{**}{x}$，最大値を $\overset{**}{f}$ とする．一般に最大点は複数あることが考えられるので，最大点の集合とその関数値の集合をそれぞれ $\overset{**}{X}$，$\overset{**}{F}$ で表す．

2.2 記　　法

本論では，次の (N1)～(N4) の記法を用いる．

(N1) 有限個の要素からなる集合 S の，j 番目の要素，要素の個数および添字集合を，それぞれ，S_j，$|S|$，$I(S)$ で表す．

(N2) アルゴリズムにおいて，集合 A へ複数の要素 $(a_1, a_2, \ldots, a_n)$ を設定するとき

$A \leftarrow (a_1, a_2, \ldots, a_n)$;

で表記する．入力変数 $(iv_1, iv_2, \ldots, iv_p)$ を与えて，$Algo\_name$ で識別されるアルゴリズムを適用し，出力変数 $(ov_1, ov_2, \ldots, ov_q)$ が得られたとき

$(ov_1, ov_2, \ldots, ov_q) \leftarrow Algo\_name(iv_1, iv_2, \ldots, iv_p)$;

で表記する．

(N3) 隣接する3点 (x_{i-1}, x_i, x_{i+1}) 1組を1つの要素とし，その要素からなる集合を X^B で表す．同様に，隣接する3点での関数値 (f_{i-1}, f_i, f_{i+1}) 1組を1つの要素とし，その要素からなる集合を F^B で表す．

(N4) アルゴリズムにおいて，隣接する3点からなる変数 (x_{i-1}, x_i, x_{i+1}) 1組を1つの要素として，集合 X^B に追加する操作を

$X^B \leftarrow X^B + (x_1, x_2, x_3)$;

で表記する．また，集合 X^B の隣接する3点1組からなる i 番目の要素 X_i^B を3つの変数 (x_1, x_2, x_3) に代入する操作を

$(x_1, x_2, x_3) \leftarrow X_i^B$;

で表記する．集合 F^B に関する操作も同様に表記する．

3. 各極大点の単峰領域と単峰性領域半径

ある点が極大(小)点となるための条件については，関数の一回微分や二回微分を用いた関係式が知られているが，極大点を含む関数値間の関係は知られていない[2]．ここでは，極大点を含むための関数値間の関係を示す．

定理1 問題(P)において，点 $x_1 < x_2 < x_3$ が存在し

$$f(x_1) < f(x_2),\ f(x_2) \geq f(x_3) \tag{4}$$

のとき，開区間 (x_1, x_3) に少なくとも1つの極大点が存在する．

(証明) 問題(P)の仮定から $f(x)$ は連続で，最大点が $[x_1, x_3]$ に存在する．ここで，(x_1, x_3) に極大点が存在しないことを仮定すると，関数 f の最大点（極大点）は x_1，x_3 いずれかに存在する．この場合，$f(x_1) < f(x_2)$ より，x_3 が最大点となる．一方，$f(x_2) \geq f(x_3)$ より，$x_2 \in (x_1, x_3)$ も最大点（極大点）となり，仮定に反する． （証明終）

性質1 関数 $f(x)$ が連続なとき，極大点 $\overset{*}{x}{}^j$ の近傍である $\beta > 0$ が存在し

$$\overset{*}{x}{}^j - \beta < \forall x_1 < \forall x_2 \leq \overset{*}{x}{}^j \implies f(x_1) < f(x_2) \leq f(\overset{*}{x}{}^j) \tag{5}$$

$$\overset{*}{x}{}^j \leq \forall x_1 < \forall x_2 < \overset{*}{x}{}^j + \beta \implies f(\overset{*}{x}{}^j) \geq f(x_1) > f(x_2) \tag{6}$$

が成立する．

(証明) 仮定1と注意2より，$\overset{*}{x}{}^j$ に十分近い点 $x_1 < \overset{*}{x}{}^j$ をとれば，区間 $(x_1, \overset{*}{x}{}^j)$ に極大点 $\overset{*}{x}{}^j$ は存在せず，しかも $f(x_1) < f(\overset{*}{x}{}^j)$ となる．f は連続だから，十分小さい $0 < \gamma < \overset{*}{x}{}^j - x_1$ をとれば，任意の $x \in (\overset{*}{x}{}^j - \gamma, \overset{*}{x}{}^j)$ に対して $f(x_1) < f(x)$ とできる．この γ に対し，式(5)が成立しないと仮定すると，$\overset{*}{x}{}^j - \gamma < x_2 < x_3 < \overset{*}{x}{}^j$ なる x_2, x_3 が存在して，$f(x_2) \geq f(x_3)$ となる．以上から，$x_1 < x_2 < x_3$ について $f(x_1) < f(x_2) \geq f(x_3)$ が成立し，定理1より区間 $(x_1, x_3) \subset (x_1, \overset{*}{x}{}^j)$ に極大点が存在する．これは，極大点に対する仮定1に反する．式(6)に関しても同様に証明できる． （証明終）

このことは，極大点の左側には単調増加領域が，右側には単調減少領域が存在することを示している．

D 上で唯一の最大点 $\overset{**}{x}$ を持つ単峰関数は，以下のように定義される[2]．

$$\begin{cases} \forall x_1 < \forall x_2 \leq \overset{**}{x} \implies f(x_1) < f(x_2) \\ \overset{**}{x} \leq \forall x_1 < \forall x_2 \implies f(x_1) > f(x_2) \\ \text{where},\ x_1 \in D,\ x_2 \in D. \end{cases} \tag{7}$$

この定義と性質1から，複数の極大点を有する多峰性関数において，各極大点における単峰領域を次のように定義できる．

定義1 極大点 $\overset{*}{x}{}^j$ の単峰領域： $[\overset{*}{a}{}^j, \overset{*}{b}{}^j] = Du(\overset{*}{x}{}^j)$

$$\begin{cases} \forall x_1 < \forall x_2 \leq \overset{*}{x}{}^j \implies f(x_1) < f(x_2) \\ \overset{*}{x}{}^j \leq \forall x_1 < \forall x_2 \implies f(x_1) > f(x_2) \\ \text{where},\ x_1 \in [a_j,\ b_j],\ x_2 \in [a_j,\ b_j] \end{cases} \tag{8}$$

を満足する最大閉区間 $[\overset{*}{a}{}^j, \overset{*}{b}{}^j]$ を $\overset{*}{x}{}^j$ の単峰領域と呼び，$[\overset{*}{a}{}^j, \overset{*}{b}{}^j] = Du(\overset{*}{x}{}^j)$ で表す．次に，各極大点における単峰領域半径を以下のように定義する．

定義2 極大点 $\overset{*}{x}{}^j$ の単峰領域半径： $r(\overset{*}{x}{}^j)$

単峰領域 $[\overset{*}{a}{}^j, \overset{*}{b}{}^j] = Du(\overset{*}{x}{}^j)$ の上下限と $\overset{*}{x}{}^j$ の距離の小さい方を $\overset{*}{x}{}^j$ における単峰領域半径 $r(\overset{*}{x}{}^j)$ と呼び

$$r(\overset{*}{x}{}^j) = \min\{\, \overset{*}{x}{}^j - \overset{*}{a}{}^j,\ \overset{*}{b}{}^j - \overset{*}{x}{}^j \,\} \tag{9}$$

となる．単峰領域と単峰領域半径の例を**図1**に示す．

(注意3) 式(1)の関数の定義より，$\overset{*}{x}{}^j = a$ の場合は $r(\overset{*}{x}{}^j) = \overset{*}{b}{}^j - \overset{*}{x}{}^j$，$\overset{*}{x}{}^j = b$ の場合は $r(\overset{*}{x}{}^j) = \overset{*}{a}{}^j - \overset{*}{x}{}^j$ となる．

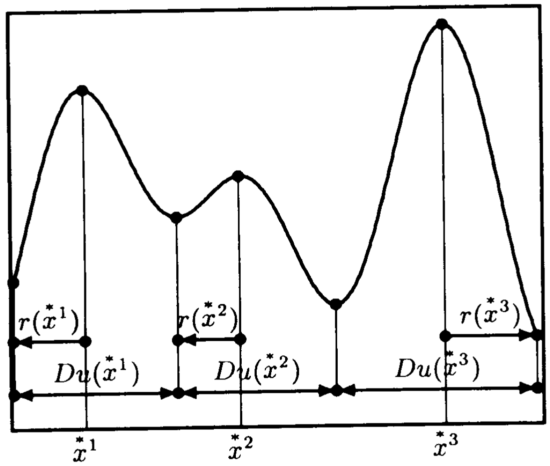


図 1 単峰領域と単峰領域半径

Fig. 1 The unimodal regions and the radius of unimodal regions.

4. 複数極大点検出アルゴリズムと検出条件

4.1 極大点検出アルゴリズム

単峰領域上にある 3 点とその関数値に関して，定理 1 から次の系が成立する．

系 1 単峰領域 $[\overset{*}{a}^j, \overset{*}{b}^j] = Du(\overset{*}{x}^j)$ 上に 3 点 $x_1 < x_2 < x_3$ が存在し，その関数値 f_1, f_2, f_3 が与えられたとき，$\overset{*}{x}^j \in (x_1, x_3)$ となるための十分条件は

$$f_1 < f_2,\ f_2 \geq f_3 \tag{10}$$

となる．

ここで，区間 $[a, b]$ 上に間隔 h で並ぶ $N+3$ 個の点列 $x_i \equiv a+ih\ (i=-1, 0, \ldots, N+1)$，$N=(b-a)/h$ と各点で関数値 $f_i \equiv f(x_i)\ (i=-1, 0, \ldots, N+1)$ が与えられたとする（今後，特に断わらない限り，点列 $x_i\ (i=-1, 0, \ldots, N+1)$ は等間隔に並んでいるものとする）．このとき，各関数値 f_i の計算は関数 $f(x)$ の定義式 (1) より

$$\begin{cases} f_i = f_o(x_i),\ x_i = a + ih,\ 0 \leq i \leq N \\ f_{-1} = f_o(a) - h = f_0 - h \\ f_{N+1} = f_o(b) - h = f_N - h \end{cases} \tag{11}$$

として求めればよい．

本節で述べるアルゴリズムは，点列 $x_i\ (i=-1, 0, \ldots, N+1)$ とその関数値 $f_i\ (i=-1, 0, \ldots, N+1)$ が与えられたとき，系 1 の関数値間の関係を利用して隣接 3 点の両端点が極大点を含むような区間を検出することを目的としている．このとき，隣接 3 点での関数値間の関係から系 1 の判定を適用して極大点を含む区間を検出する条件は

$$\begin{cases} f_{i-1} < f_i,\ f_i \geq f_{i+1} \\ \qquad \Longrightarrow \overset{*}{x}^j \in (x_{i-1}, x_{i+1}), \\ \qquad\quad i = 0, 1, \ldots, N \end{cases} \tag{12}$$

となる．

以上から，次のような複数極大点を含む区間を検出するアルゴリズムを構成することができる．

Algorithm1 $(X^B, F^B) \leftarrow Bloptuv(f_o, a, b, h)$;

[Input] f_o : 最大化関数，a, b : x の上下限，h : 標本点の間隔 $(h > 0)$.

[Output] X^B : 極大点を含む隣接 3 点の集合，F^B : 極大点を含む隣接 3 点の関数値の集合．

Step.1 (Initialize and calculate the number of division)

$X^B \leftarrow \emptyset$; $F^B \leftarrow \emptyset$; $N \leftarrow (b-a)/h$;

Step.2 (Detect the neighbour three points such that these two end points include a local maximum and add three points and these functional values to related two sets)

for $i \leftarrow 0$ **to** N **do**
  $x_i \leftarrow a + ih$; $f_i \leftarrow f_o(x_i)$;
  if $i = 0$ **then** $f_{-1} \leftarrow f_0 - h$;
  if $i = N$ **then** $f_{N+1} \leftarrow f_N - h$;
  if $f_{i-1} < f_i$ **and** $f_i \geq f_{i+1}$ **then**
    $X^B \leftarrow X^B + (x_{i-1}, x_i, x_{i+1})$;
    $F^B \leftarrow F^B + (f_{i-1}, f_i, f_{i+1})$;
  fi;
od;

このアルゴリズムは単純で容易に構成可能であり，計算量は $O(N)$ となる．

4.2 最大点とすべての極大点を含む隣接 3 点を検出する条件

Algorithm1 で利用した，系 1 の隣接 3 点での関数値間の関係は極大点を含むための十分条件であるため，条件 (12) が成立しなくても隣接 3 点の両端点が極大点 $\overset{*}{x}^j$ を含む場合があり，検出に失敗する恐れがある．そこで，隣接 3 点の両端点が極大点を含む区間を必ず検出する条件を以下に与える．

命題 1 極大点 $\overset{*}{x}^j$ に対して，隣接 4 点が $x_{i-1} < x_i \leq \overset{*}{x}^j < x_{i+1} < x_{i+2}$ を満足し，これらの 4 点が単峰領域 $Du(\overset{*}{x}^j)$ に含まれているものとする．このとき，Algorithm1 は，条件 (12) により隣接 3 点 (x_{i-1}, x_i, x_{i+1})，(x_i, x_{i+1}, x_{i+2}) のいずれか一方で，

極大点 $\overset{*}{x}^j$ を検出できる．

(証明) 隣接4点が $x_{i-1} < x_i \leq \overset{*}{x}^j < x_{i+1} < x_{i+2}$ で単峰領域にあるから，条件 (12) より，$f_i < f_{i+1}$ のときは $f_{i-1} < f_i < f_{i+1} \geq f_{i+2}$ となり (x_i, x_{i+1}, x_{i+2}) で，$f_i \geq f_{i+1}$ のときは，$f_{i-1} < f_i \geq f_{i+1} > f_{i+2}$ となり (x_{i-1}, x_i, x_{i+1}) で極大点 $\overset{*}{x}^j$ を検出できる．（証明終）

極大点検出アルゴリズムで極大点を検出するために，標本点の間隔が満足する条件を以下に示す．

命題2 Algorithm1 において，極大点 $\overset{*}{x}^j$ を検出するために標本点の間隔 h が満足する条件は，次式で与えられる．

$$h \leq \frac{1}{2} r(\overset{*}{x}^j) . \qquad (13)$$

(証明)

(1) $\overset{*}{x}^j = a$ の場合は，3点 $(x_{-1} < x_0 = \overset{*}{x}^j < x_1)$ が単峰領域上にあれば，$f_{-1} < f_0 = f(\overset{*}{x}^j) > f_1$ となり，極大点を検出できる．式 (1) より，x_{-1} はつねに単峰領域上に存在するから，$x_1 - x_0 \leq r(\overset{*}{x}^j)$ が成立すればよいので条件 (13) で十分極大点を検出できる．
(2) $\overset{*}{x}^j = b$ の場合も，1) と同様に示すことができる．
(3) $\overset{*}{x}^j \neq a$, $\overset{*}{x}^j \neq b$ の場合，極大点 $\overset{*}{x}^j$ を検出するための4点の条件は，命題1より，単峰領域 $[\overset{*}{a}^j, \overset{*}{b}^j] = Du(\overset{*}{x}^j)$ 上にある i と点 $x_{i-1} < x_i < x_{i+1} < x_{i+2}$ が存在して

$$\left\{ \begin{array}{l} \overset{*}{a}^j \leq x_{i-1}\,, \quad x_{i+2} \leq \overset{*}{b}^j \\ x_i \leq \overset{*}{x}^j < x_{i+1}\,, \quad i = 0, 1, \ldots, N \end{array} \right. \qquad (14)$$

となる．ここで式 (14) を満足しながら，可能な限り h を大きくすることを考える．この場合，区間 $[\overset{*}{a}^j, \overset{*}{x}^j]$ に2点 x_{i-1}, x_i の点だけが入り，区間 $(\overset{*}{x}^j, \overset{*}{b}^j]$ に2点 x_{i+1}, x_{i+2} の点だけが入ればよい（**図2**）から

$$\left\{ \begin{array}{l} x_{i-2} \leq \overset{*}{a}^j, \;\; \overset{*}{a}^j < x_{i-1} < x_i \leq \overset{*}{x}^j \\ \overset{*}{x}^j < x_{i+1} < x_{i+2} \leq \overset{*}{b}^j, \;\; \overset{*}{b}^j < x_{i+3} \end{array} \right. \qquad (15)$$

となる．ここで，$x_i = x_{i-1} + h$, $x_{i+2} = x_{i+1} + h$ であるから，$\overset{*}{a}^j$ と x_{i-1} の間隔を s_1，$\overset{*}{x}^j$ と x_{i+1} の間隔を s_2 とすると

$$\left\{ \begin{array}{ll} s_1 + h \leq \overset{*}{x}^j - \overset{*}{a}^j, & 0 \leq s_1 \leq h \\ s_2 + h \leq \overset{*}{b}^j - \overset{*}{x}^j, & 0 < s_2 \leq h \end{array} \right. \qquad (16)$$

となる．s_1，s_2 の値にかかわらずつねに式 (16) が成立するためには，単峰領域半径の定義 (9) と $0 \leq s_1 \leq h$, $0 < s_2 \leq h$ より

$$2h \leq r(\overset{*}{x}^j) \qquad (17)$$

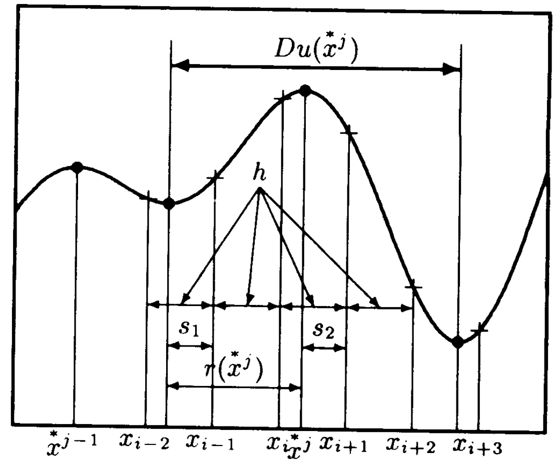


図2 極大点 $\overset{*}{x}^j$ での単峰領域上の等間隔 h の4点
Fig. 2 Four points with the same interval h on the unimodal regions of a local maximum.

となり，式 (13) が成立する． （証明終）

命題2から，Algorithm1 ですべての極大点を検出するために標本点の間隔が満足する条件は，すべての極大点の単峰領域半径に関して命題2と同様に成立すればよいから，次の命題3を示すことができる．

命題3 Algorithm1 において，すべての極大点を検出するために標本点の間隔 h が満足する条件は，次式で与えられる．

$$h \leq \min_{i \in I(\overset{*}{X})} \left\{ \frac{1}{2} r(\overset{*}{X}_i) \right\} . \qquad (18)$$

なお，最大点のみを求めるには

$$h \leq \min_{i \in I(\overset{**}{X})} \left\{ \frac{1}{2} r(\overset{**}{X}_i) \right\} \qquad (19)$$

を満足すればよい．

以上から，命題3または式 (19) の条件を満足するように h を定めれば，Algorithm1 はすべての極大点または最大点を必ず検出できることが分かる．

5. 複数極大点探索アルゴリズムと最大点探索アルゴリズム

5.1 複数極大点探索アルゴリズム

本節では，複数極大点検出アルゴリズムで得られた極大点を含む隣接3点の集合とその関数値の集合から，与えられた要求精度で複数極大点を求めるアルゴリズムを示す．このアルゴリズムは局所最適化手法が与えられていれば，各極大点を含む隣接3点に対して，局所最適化手法を適用するだけでよいから，非常に容易に実現できる．また，極大点を含む3点がすでに得られているので，局所最適化手法で極大点を囲い込むステップが不要である．そこで，局所最適化手法内部で極小点を囲い込むステップを省くことが可能であれば，

その関数評価回数を減少させることができる．ここでは，このような考えに基づく3点での二次補間を用いた局所最適化手法のアルゴリズムの概要を示し（詳細は文献 2)，4) 等を参照)，次に，局所最適化手法を用いた複数極大点探索手法のアルゴリズムを示す．

極大点を含む単峰領域上にある隣接3点から，極大点 $\overset{*}{x}$ とその関数値 $\overset{*}{f}$ を要求精度 ϵ_x または ϵ_f で求める局所最適化アルゴリズムの概要は以下のようになる．

Algorithm2 $(\overset{*}{x}, \overset{*}{f}) \leftarrow Loptuv(f_o, X_i^B, F_i^B, \epsilon_x, \epsilon_f)$;

[Input] f_o：最大化関数，
X_i^B：極大点を囲む隣接3点，
F_i^B：極大点を囲む隣接3点の関数値，
ϵ_x：x の要求精度，
ϵ_f：関数値の要求精度．

[Output] $\overset{*}{x}$：極大点，$\overset{*}{f}$：極大点での関数値．

Step.1 (Make repeat until the stop condition is satisfied)
repeat do

Step.2 (Find an approximated local maximum $\overset{*}{x}$ by quadratic interpolation, and check the stop condition)
$(x_1, x_2, x_3) \leftarrow X_i^B$; $(f_1, f_2, f_3) \leftarrow F_i^B$;
$w_1 \leftarrow \{(x_2)^2 - (x_3)^2\}f_1 + \{(x_3)^2 - (x_1)^2\}f_2 + \{(x_1)^2 - (x_2)^2\}f_3$;
$w_2 \leftarrow (x_2 - x_3)f_1 + (x_3 - x_1)f_2 + (x_1 - x_2)f_3$;
$\overset{*}{x} \leftarrow w_1/2w_2$; $\overset{*}{f} \leftarrow f_o(\overset{*}{x})$;
if $|\overset{*}{x} - x_2| < |1 + \overset{*}{x}|\epsilon_x$ **or** $|\overset{*}{f} - f_2| < |1 + \overset{*}{f}|\epsilon_f$
then break;

Step.3 (Find new reduced points (x_1, x_2, x_3) and these functional value (f_1, f_2, f_3) from $(x_1, x_2, x_3, \overset{*}{x})$ such that end two points of new three points include a local maximum)
od;

このアルゴリズムの収束次数は約1.3次（超一次収束）である[4)]ため，効率の良いアルゴリズムであると考えられる．

この Algorithm2 を用いて，複数極大点の集合とその関数値の集合を与えられた要求精度で求めるアルゴリズムは，極大点を含む隣接3点の集合の要素数だけ局所最適化手法を適用し，得られた極大点とその関数値を，それぞれの集合に加えることにより構成できる．よって，極大点の集合 $\overset{*}{X}$ とその関数値の集合 $\overset{*}{F}$ を要求精度 ϵ_x または ϵ_f で求めるアルゴリズムは以下のようになる．

Algorithm3 $(\overset{*}{X}, \overset{*}{F}) \leftarrow Mloptuv(f_o, a, b, h, \epsilon_x, \epsilon_f)$;

[Input] f_o：最大化関数， a, b：x の上下限，
h：標本点の間隔 $(h > 0)$，
ϵ_x：x の要求精度，
ϵ_f：関数値の要求精度．

[Output] $\overset{*}{X}$：極大点の集合，
$\overset{*}{F}$：極大点での関数値の集合．

Step.1 (Initialize)
$\overset{*}{X} \leftarrow \emptyset$; $\overset{*}{F} \leftarrow \emptyset$;

Step.2 (Detect the set of three points such that these end two points include a local maximum)
$(X^B, F^B) \leftarrow Bloptuv(f_o, a, b, h)$;

Step.3 (Apply the local maximizer to each detected three neighbouring points and these functional values)
for $i \leftarrow 1$ **to** $|X^B|$ **do**
$(\overset{*}{x}, \overset{*}{f}) \leftarrow Loptuv(f_o, X_i^B, F_i^B, \epsilon_x, \epsilon_f)$;
$\overset{*}{X} \leftarrow \overset{*}{X} + \{\overset{*}{x}\}$; $\overset{*}{F} \leftarrow \overset{*}{F} + \{\overset{*}{f}\}$;
od;

このアルゴリズムの計算量は，*Loptuv* の平均計算量を $\overline{L}$ とすると，$O(N + |X^B| \cdot \overline{L})$ となる．

5.2 最大点探索アルゴリズム

Algorithm3 では，最大点を探索する観点からすると，最大点と異なる他の極大点に対してもすべて局所最適化手法を適用するため，効率が悪い．従来，最大点を求める決定論的な手法では，得られている点列とその関数値から Lipschitz 条件を用いて極大点を含む可能性のない領域を除くことによって，最大点を含む領域を縮小していく．しかし，この手法ではあらかじめ Lipschitz 定数を与える必要があり，この定数の適切な与え方が一般には困難である．そこで本節では，極大点を含む隣接3点とその関数値から最大値が得られる可能性のない隣接3点を取り除くことにより，局所最適化手法を適用する回数を減少させる手法について述べる．まず，極大点を含む隣接3点とその関数値から，3点の両端区間での関数値の上限を与える定理を示す．

定理2 単峰領域 $Du(\overset{*}{x}^j)$ 上に3点 $(x_1 < x_2 < x_3)$ が存在し，その関数値 (f_1, f_2, f_3) が与えられたとする．このとき，区間 $[x_1, x_3]$ で関数 f が凹関数で，$\overset{*}{x}^j \in [x_1, x_3]$ を満足するならば

$$\begin{cases} f(\overset{*}{x}{}^{j}) \leq \max\{f_{ub1},\ f_{ub2}\} \\ f_{ub1} = \dfrac{x_3 - x_2}{x_2 - x_1}(f_2 - f_1) + f_2 \\ f_{ub2} = \dfrac{x_2 - x_1}{x_3 - x_2}(f_2 - f_3) + f_2 \end{cases} \tag{20}$$

となる.

(証明) $x \in [x_1, x_2]$ なら, $x \leq x_2 < x_3$ と f の凹性より

$$\begin{cases} f_2 \geq \theta f_3 + (1-\theta) f(x) \\ \text{where}\ ,\ \theta = (x_2 - x)/(x_3 - x) \end{cases} \tag{21}$$

これを整理して

$$\begin{aligned} f(x) &\leq \frac{f_2}{1-\theta} - \frac{\theta f_3}{1-\theta} = \frac{\theta}{1-\theta}(f_2 - f_3) + f_2 \\ &= \frac{f_2 - f_3}{x_3 - x_2}(x_2 - x) + f_2 \equiv g_1(x)\ . \end{aligned} \tag{22}$$

同様に $x \in [x_2,\ x_3]$ なら

$$f(x) \leq \frac{f_2 - f_1}{x_2 - x_1}(x - x_2) + f_2 \equiv g_2(x)\ . \tag{23}$$

ゆえに

$$\begin{aligned} &\max_{x_1 \leq x \leq x_3} f(x) \\ &\leq \max\{ \max_{x_1 \leq x \leq x_2} g_1(x), \max_{x_2 \leq x \leq x_3} g_2(x)\} \\ &= \max\{f_2, g_1(x_1), g_2(x_3)\} \\ &= \max\{f_2, f_{ub2}, f_{ub1}\}\ . \end{aligned} \tag{24}$$

ここで, $f_2 > \max\{\ f_{ub1}, f_{ub2}\ \}$ とすると

$$\begin{cases} f_{ub1} - f_2 = \dfrac{x_3 - x_2}{x_2 - x_1}(f_2 - f_1) < 0 \\ f_{ub2} - f_2 = \dfrac{x_2 - x_1}{x_3 - x_2}(f_2 - f_3) < 0 \end{cases} \tag{25}$$

より, $f_2 < f_1$, $f_2 < f_3$ となり, f の凹性に反する. ゆえに $f_2 \leq \max\{\ f_{ub1}, f_{ub2}\ \}$ となり, 式 (24) から

$$\max_{x_1 \leq x \leq x_3} f(x) = f(\overset{*}{x}{}^{j}) \leq \max\{f_{ub2}, f_{ub1}\} \tag{26}$$

が得られる. (証明終)

この定理から, 現在得られている最大値を $\overset{**}{f}$ とし, 最大値を誤差 ϵ_f で求めること考えたとき

$$\begin{cases} \overset{**}{f} > (1+\epsilon_f)\max\{f_{ub1}, f_{ub2}\} \\ \qquad f_{ub1} = \dfrac{x_3 - x_2}{x_2 - x_1}(f_2 - f_1) + f_2 \\ \qquad f_{ub2} = \dfrac{x_2 - x_1}{x_3 - x_2}(f_2 - f_3) + f_2 \end{cases} \tag{27}$$

であれば, この3点に局所最適化手法を適用して得られる極大値は現在の最大値を超えることがないので, この3点は除くことができる. また, Algorithm1 で得られた隣接3点が等間隔で並んでいることに注意すると, 式 (27) は, 次のように簡単になる.

$$\begin{cases} f(\overset{*}{x}{}^{j}) < (1+\epsilon_f)\max\{f_{ub1},\ f_{ub2}\} \\ f_{ub1} = 2f_2 - f_1\ ,\ \ f_{ub2} = 2f_2 - f_3 \end{cases} \tag{28}$$

最大点探索アルゴリズムの中で, 与えられた要求精度で極大点を求めるために局所最適化手法を用いる. そこで, 局所最適化手法の引数に現在得られている最大値 $\overset{**}{f}$ を加え, このアルゴリズム内で要求精度で解が得られたかを判定すると同時に式 (27) の判定を加え, 式 (27) が成立したときには最大点が得られる可能性がないので, 繰返しを途中で打ち切ることにより, より効率的なアルゴリズムを構成することができる. このようなアルゴリズムの概要を以下に示す.

Algorithm4

$(\overset{*}{x}, \overset{*}{f}, mf) \leftarrow Loptuv1(f_o, X_i^B, F_i^B, \epsilon_x, \epsilon_f, \overset{**}{f})$;

[Input] f_o : 最大化関数,
X_i^B : 極大点を囲む隣接 3 点,
F_i^B : 極大点を囲む隣接 3 点の関数値,
ϵ_x : x の要求精度,
ϵ_f : 関数値の要求精度,
$\overset{**}{f}$: 現在得られている最大値.

[Output] $\overset{*}{x}$: 極大点 , $\overset{*}{f}$: 極大点での関数値,
mf : 極大点が得られた ($= \boldsymbol{true}$) か, 途中で打ち切った ($= \boldsymbol{false}$) を判定するフラグ.

Step.1 (Repeat until a local maximum is found or inequality (27) is satisfied)
repeat do

Step.2 (Find an approximated local maximum $\overset{*}{x}$ and it's functional value $\overset{*}{f}$ by quadratic interpolation)

Step.3 (If a local maximum is found, then set mf and break iteration)
if $|\ \overset{*}{x} - x_2| < |1+\overset{*}{x}\ |\epsilon_x$ **or** $|\ \overset{*}{f} - f_2| < |1+\overset{*}{f}\ |\epsilon_f$
then $mf \leftarrow \boldsymbol{true}$; **break**; **fi**;

Step.4 (Find reduced points (x_1, x_2, x_3) and functional values (f_1, f_2, f_3))

Step.5 (If inequality (27) is satisfied then set mf, and break iteration)
$f_{ub1} \leftarrow (x_3 - x_2)(f_2 - f_1)/(x_2 - x_1) + f_2$;
$f_{ub2} \leftarrow (x_2 - x_1)(f_2 - f_3)/(x_3 - x_2) + f_2$;
if $\overset{**}{f} > (1+\epsilon_f)\max\{\ f_{ub1}, f_{ub2}\ \}$
then $mf \leftarrow \boldsymbol{false}$; **break**; **fi**;
od;

以上から，複数極大点検出アルゴリズム（Algorithm1）と局所最適化アルゴリズム（Algorithm4）を用いて，最大点の集合 $\overset{**}{X}$ とその関数値の集合 $\overset{**}{F}$ を与えられた要求精度で求めるアルゴリズムは以下のように構成できる．

Algorithm 5 $(\overset{**}{X}, \overset{**}{F}) \leftarrow Gloptuv(f_o, a, b, h, \epsilon_x, \epsilon_f)$;

[Input] f_o：最大化する関数，
a, b：x の上下限，
h：標本点の間隔 ($h > 0$)，
ϵ_x：x の要求精度，
ϵ_f： 関数値の要求精度．

[Output] $\overset{**}{X}$：最大点の集合，
$\overset{**}{F}$：最大点での関数値の集合．

[Notation] $Imax(F^B)$：集合 F^B の最大値を与える要素の添字を返す関数．

Step.1 (Initialize)
$\overset{**}{X} \leftarrow \emptyset$; $\overset{**}{F} \leftarrow \emptyset$; $\overset{**}{f} \leftarrow -\infty$;

Step.2 (Detect the set of three points such that these two end points include a local maximum)
$(X^B, F^B) \leftarrow Bloptuv(f_o, a, b, h)$;
while $X^B \neq \emptyset$ **do**

Step.3 (Apply the local maximizer to giving each detected three neighbouring points with the maximal value in F^B)
$m \leftarrow Imax(F^B)$;
$(\overset{*}{x}, \overset{*}{f}, mf) \leftarrow Loptuv1(f_o, X^B_m, F^B_m, \epsilon_x, \epsilon_f, \overset{**}{f})$;

Step.4 (If a local maximum is found, then add the current local maximum and it's functional value to related sets)
if $mf = \boldsymbol{true}$ **then**
$\overset{**}{X} \leftarrow \overset{**}{X} + \{\overset{*}{x}\}$; $\overset{**}{F} \leftarrow \overset{**}{F} + \{\overset{*}{f}\}$; **fi**;

Step.5 (Remove the three points and these functional values such that these values cannot be greater than the current maximal value)
if $\overset{*}{f} > \overset{**}{f}$ **then**
$\overset{**}{f} \leftarrow \overset{*}{f}$;
for $i \leftarrow 1$ **to** $|X^B|$ **do**
$(f_1, f_2, f_3) \leftarrow F^B_i$;
$f_{ub} \leftarrow (1+\epsilon_f)\max\{2f_2 - f_1, 2f_2 - f_3\}$;
if $f_{ub} < \overset{**}{f}$ **then**
$X^B \leftarrow X^B - \{X^B_i\}$;
$F^B \leftarrow F^B - \{F^B_i\}$; **fi**;
od;
fi;
fi;
od;

Step.6 (Find the true set of global maxima and these functional values)
for $i \leftarrow 1$ **to** $|\overset{**}{X}|$ **do**
if $\overset{**}{F}_i < (1-\epsilon_f)\overset{**}{f}$ **then**
$\overset{**}{X} \leftarrow \overset{**}{X} - \{\overset{**}{X}_i\}$; $\overset{**}{F} \leftarrow \overset{**}{F} - \{\overset{**}{F}_i\}$; **fi**;
od;

（注意 4） このアルゴリズムにおいて，複数極大点アルゴリズム（*Bloptuv*）で検出された隣接 3 点の両端区間上で関数の凹性が成立しない場合は，局所最適化アルゴリズム（*Loptuv*1）の中や Step.5 で，最大点を含む隣接 3 点を取り除いてしまう危険性がある．

6. 数値例と考察

文献 13) で示されている次の標準的な一変数最小化テスト関数を最大化テスト関数に置き換えて，提案した 2 手法に対する数値実験を行った．なお，f_5, f_6 の係数を**表 1** に示す．

$$f_1(x) = -\sin(x) - \sin(\tfrac{10x}{3}) - \ln(x) + 0.84x,$$
$x \in [2.7, 7.5]$; 3 極大点（1 最大点）

$$f_2(x) = -\sin(x) - \sin(\tfrac{2x}{3}),$$
$x \in [3.1, 20.4]$; 3 極大点（1 最大点）

$$f_3(x) = \sum_{i=1}^{5} i \sin((i+1)x + i),$$
$x \in [-10, 10]$; 19 極大点（3 最大点）

$$f_4(x) = -(x + \sin(x))e^{-x^2},$$
$x \in [-10, 10]$; 1 極大点

$$f_5(x) = \sum_{i=1}^{10} \frac{1}{(k_i^1(x - a_i^1))^2 + c_i^1},$$
$x \in [0, 10]$; 9 極大点（1 最大点）

$$f_6(x) = \sum_{i=1}^{10} \frac{1}{(k_i^2(x - a_i^2))^2 + c_i^2},$$
$x \in [0, 10]$; 6 極大点（1 最大点）

数値実験での条件を以下に示す．

（1） 計算機：NEC EWS-4800/230
（2） 浮動小数形式と計算精度：IEEE754 規格の倍精度
（3） 要求精度：$\epsilon_x = \epsilon_f = 10^{-6}$
（4） 関数 $f_1 \sim f_4$ での h の設定：周期関数 $\sin(\cdot)$ の最も短い周期の 1/4 に設定
（5） 関数 $f_5 \sim f_6$ での h の設定：次の (a)～(b) の仮定をし，

表 1　テスト関数 $f_5(x)$ と $f_6(x)$ の係数
Table 1　Coefficients of the test functions $f_5(x)$ and $f_6(x)$.

i	a_i^1	k_i^1	c_i^1	a_i^2	k_i^2	c_i^2
1	3.040	2.983	0.192	4.696	2.871	0.149
2	1.098	2.378	1.140	4.885	2.328	0.166
3	0.674	2.439	0.127	0.800	1.111	0.175
4	3.537	1.168	0.132	4.986	1.263	0.183
5	6.173	2.406	0.125	3.901	2.399	0.128
6	8.679	1.236	0.189	2.395	2.629	0.117
7	4.503	2.868	0.187	0.945	2.853	0.115
8	3.328	1.378	0.171	8.371	2.344	0.148
9	6.937	2.348	0.188	6.181	2.592	0.188
10	0.700	2.268	0.176	5.713	2.929	0.198

(a)　極大点の上限 m の中で α 割の極大点が最大点を与える候補点 $\overset{*}{x}{}^j\ (j=1,2,\ldots,m\alpha)$ とする

(b)　各候補点の単峰領域が次式
$\overset{*}{b}{}^j-\overset{*}{a}{}^j=(b-a)/(m\alpha)$
のように等しく，かつ
$\overset{*}{b}{}^j-\overset{*}{x}{}^j=\overset{*}{x}{}^j-\overset{*}{a}{}^j=r(\overset{*}{x}{}^j)=(b-a)/(2m\alpha)$
とする

h の設定を

$$h=r(\overset{*}{x}{}^j)=(b-a)/4m\alpha$$

によって定めた．ただし，$b-a=10$, $m=10$ で，$\alpha=0.7$ とした．

文献 13) に示されている他の手法の数値例と，数値実験で得られた本提案の 2 手法（手法 1: Algorithm3 の複数極大点探索手法，手法 2: Algorithm5 の最大点探索手法）の数値例を比較したものを**表 2** に示す．なおこの表で，Žil1 [16]，Žil2 [15]，Stro は確率論的手法，Shub [11]，Brent [4]，Batis は Lipschitz 条件を用いた決定論的手法に分類される．また，決定論的手法で用いた 3 手法の Lipschitz 定数は，各目的関数の探索領域 D 上での 1 回の微係数値の上界（Shub, Batis）もしくは，2 回の微係数値の上界（Brent）を用いているので，いずれも最適値を与えている．

本提案の 2 手法は他の手法と比べて比較的効率が良く，本 2 手法と同様の決定論的な手法（Shub, Brent, Batis）と比較すると，非常に効率が良いことが分かる．その理由は，多峰性関数の各極大点を囲む隣接 3 点からなる部分領域を検出し，各部分領域に超一次収束性を持つ効率的な局所最適化手法を適用したためであると考えられる．

確率論的な手法（Žil1, Žil2, Stro）も効率が良いが，最大点を与えられた確率（Žil1, Žil2 では 0.95 に設定）で求めるため，確実に最大点が求められるという保証がない．また，より確実に最大点を求めるため

表 2　標準テスト関数における他の手法と本 2 手法の関数評価回数による比較
Table 2　A comparison by the number of function evaluations among the other methods and our two methods in the standard test functions.

関数	Žil1	Žil2	Stro	Shub	Brent	Batis
f_1	33	29	45	462	25	120
f_2	37	38	442	448	45	158
f_3	125	165	150	3817	161	816
f_4	35	34	98	376	229	83
f_5	42	41	102	280	294	484
f_6	45	44	69	624	492	325
Total	317	351	906	5907	1246	1986

関数	本手法 1(h)	本手法 2(h)
f_1	40(0.47)	18(0.47)
f_2	46(1.57)	19(1.57)
f_3	242(0.26)	96(0.26)
f_4	24(1.57)	23(1.57)
f_5	89(0.35)	37(0.35)
f_6	98(0.35)	50(0.35)
Total	539	243

要求精度は Shub 以外すべて $\epsilon_x=\epsilon_f=10^{-6}$ に設定し，Shub は他の手法と比較できる値を得るために $\epsilon_x=10^{-4}$，$\epsilon_f=10^{-6}$ に設定．

に確率を 1 に近づけると効率の低下を招く．さらに，確率論的な手法は，確率関数を現在観測した標本点から計算し，この確率関数からある評価関数を最小化して次の標本点を求めるため，1 つの標本点の生成に多大な計算時間を要し，関数評価以外のオーバヘッドが非常に大きい．

手法 2 は今回の比較の中でも最も効率が良い．これは，手法 1 と手法 2 を比較すれば分かるように，式 (27) によって最大点を見いだす可能性のない 3 点を判定し，その 3 点に対して局所最適化手法を適用しないことで，無駄な関数評価を減らしたことが効果的に働いているといえる．また，手法 1 は局所最適化手法が用意されていれば，Algorithm1 と Algorithm3 のプログラム実行部を合わせても 50〜70 行程度で実装でき，きわめて容易に実現可能であり，関数評価以外のオーバヘッドもきわめて少ない．手法 1 は，ある意味で自明なアルゴリズムであるが，h を適切に設定すれば，技巧を凝らしたアルゴリズムと十分比較しうる効率を有していることが分かる．

点の間隔 h の最適な設定は一般には難しいが，たとえばテスト関数 $f_1\sim f_4$ のように，sin 等の周期関数の項が複数極大点の存在に大きく関係し，各極大点で $\overset{*}{x}{}^j-\overset{*}{a}{}^j\cong\overset{*}{b}{}^j-\overset{*}{x}{}^j\ (i=1,2,\ldots,m)$ が成立する場合は，最も短い周期 T の 1/4 を h として与えればよい．また，h は必ずしも最適な値でなくてもよく，最適な

値よりも小さな h を設定してよい．さらに，h の設定による計算量の増加は $N=(b-a)/h$ より h に反比例して増加するだけである．一方，Lipschitz 条件を用いた手法では Lipschitz 定数の値から計算量を予測するのは困難である．h は常微分方程式の解法におけるパラメータとしてのステップ幅や DFT のサンプリング間隔と対応付けて考えることができ，Lipschitz 定数よりも直感的にも理解し易い．

7. おわりに

従来，多峰性関数に関する定義やその性質がほとんど知られていなかったが，本論では，有限個の極大点を探索領域内に有する連続な一変数多峰性関数において，極大点の左近傍に単調増加領域が右近傍に単調減少領域が必ず存在するという性質を導いた．この性質から多峰性関数の単峰領域と単峰領域半径を新たに定義した．単峰領域の定義から単峰領域上にある3隣接点が極大点を含むための十分条件を関数値間の関係として導いた．

次に，3隣接点が極大点を含むための関数値間の関係を用いて，等間隔の点列とその関数値を評価し，この点列と関数値から3点の両端点が極大点含む領域を検出する複数極大点検出アルゴリズムを構成した．本アルゴリズムはきわめて容易に実現可能で，アルゴリズムの制御パラメータは隣接点の間隔だけを指定するだけで済み，利用者のパラメータ設定の負担も軽い．

また，複数極大点検出アルゴリズムで極大点を必ず検出するための条件を，関数値間の関係として与え，このアルゴリズムが最大点を検出する条件とすべての極大点を検出する条件を，隣接点の間隔と単峰領域半径の関係式で与えた．

さらに，複数極大点検出アルゴリズムで得られた両端点が極大点を含む3点を要素とする集合を用いて次の2つのアルゴリズム：(1) 3点からなる各要素すべてに局所最適化手法を適用し，与えられた要求精度で複数極大点を求めるアルゴリズム，(2) 現在得られている最大値を用いて，隣接3点からなる各要素でこの最大値を超える可能性のない要素に対して局所最適化手法を適用しない等により，効率的に最大点を求めるアルゴリズムを構成した．複数極大点探索アルゴリズムは局所最適化手法がすでに実現されていれば，非常に容易に実現可能である．

この2つのアルゴリズムを6つのテスト関数で数値実験を行い，他のアルゴリズムの同様の数値例と比較した結果，複数極大点探索アルゴリズムは比較的効率が良く，最大点探索アルゴリズムは最も効率が良いことが示された．また，信頼性の高い決定論的なアルゴリズムの中では両者とも他の手法と比べ非常に効率が良いことが示された．

今後の課題として，(1) Lipshcitz 定数を点列とその関数値から適応的に推定し，その定数を用いてより効率的なアルゴリズムの構成を試みる，(2) 多変数多峰性関数へ適用できるようにアルゴリズムを拡張する，ことがあげられる．

謝辞 論文の構成や証明等に貴重なコメントいただいた査読者の方々に，深く感謝いたします．

参考文献

1) Aluffi-Penttini, A.V., Parisi, V. and Zirilli, F.: Global Optimization and Stochastic Differential Equations, *J. Optimization Theory and Applications*, Vol.47, No.1, pp.1–16 (1985).
2) Avriel, M.: *Nonlinear Programming – Analysis and Methods*, pp.225–226, Prentice-Hall (1976).
3) Baritompa, W.: Customizing Methods for Global Optimization – A Geometric View Point, *J. Global Optimization*, Vol.3, No.2, pp.193–212 (1993).
4) Brent, R.P.: *Algorithms for Minimization without Derivatives*, Prentice-Hall, New Jersey (1973).
5) 藤井康雄，市田浩三，清野　武：区間演算を利用した多変数関数の最大値探索法，情報処理，Vol.18, No.11, pp.1095–1101 (1977).
6) Goldberg, D.E.: *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley (1989).
7) Hansen, E.R.: Global Optimization Using Interval Analysis: The One-Dimensional Case, *J. Optimization Theory and Applications*, Vol.29, No.3, pp.331–344 (1979).
8) Kirkpatrick, S., Gelatt, C.D., Jr. and Vecchi, M.P.: Optimization by Simulated Annealing, *Science*, Vol.220, No.4598, pp.671–680 (1983).
9) 水野眞治：分枝限定法をもちいた方程式の解法と関数の最小化，*J. of the Operations Research Society of Japan*, Vol.30, No.1, pp.41–58(1987).
10) Mockus, J.: *Bayesian Approach to Global Optimization – Theory and Applications*, Kluwer Academic Publishers (1989).
11) Shubert, B.O.: A Sequential Method Seeking the Global Maximum of a Function, *SIAM J. Num. Anal.*, Vol.9, No.3, pp.379–388 (1972).
12) 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，*J. of the Operations Research Society of Japan*, Vol.19, No.4, pp.295–307 (1987).
13) Törn, A.A. and Žilinskas, A.: *Global Optimization, Lecture Notes in Computer Science*,

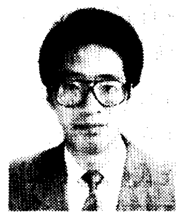
vol.350, pp.176–182, Springer-Verlag (1989).

14) Zhiglijavsky, A.A.: *Theory of Random Search*, pp.9–14, Kluwer Academic Publishers (1991).

15) Žilinskas, A.: Optimization of One-Dimensional Multimodal Functions, Algorithm AS 133, *Applied Statistics*, Vol.23, pp.367–375 (1978).

16) Žilinskas, A.: Two Algorithms for One-dimensional Multimodal Minimization, *Math. Operationsforsch. Stat. 12, Ser. Optimization*, Vol.12, No.1, pp.53–63 (1981).

(平成 7 年 7 月 27 日受付)
(平成 8 年 6 月 6 日採録)



金光　秀雄（正会員）

昭和 55 年北海道大学工学部電気工学科卒業．昭和 57 年同大学大学院修士課程修了．同年，松下電器産業（株）入社．現在，北海道教育大学教育学部函館校助教授．非線形最適化手法の研究に従事．電子情報通信学会，IEEE，ACM 等各会員．

新保　　勝（正会員）

昭和 36 年東京大学工学部応用物理学科（数理工学）卒業．昭和 41 年同大学大学院博士課程単位取得退学．工学博士．現在，北海道大学教授．米国コーネル大学客員教授，英国サイエンスリサーチカウンシル上級客員研究員．この間主として心理物理認識の研究に従事．電子情報通信学会，日本音響学会，テンソル学会，日本材料科学会等各会員．