

対応の与えられた点集合間のミニマックス近似問題について

今井 桂子

九州工業大学情報科学センター

二つの似かよった点集合の間の最適当てはめを求める問題は、パターン認識・画像処理・統計などで現われる。本稿では、1対1対応のついた平面上の n 点からなる二つの点集合 S, T が与えられたとき、 S を平行移動と回転することにより、対応する点同士の距離の最大値が最小になるような当てはめを求めるというミニマックス型の問題を考える。この問題を、 n 個の3変数関数の最大値をとる関数の最小化問題として定式化し、それに対する $O(n^2\lambda_{10}(n)\log n)$ の手間のアルゴリズムを与える。ここで $\lambda_{10}(n)$ は10次のDavenport-Schinzel列の最大長であり、 n に関してほとんど線形に近い関数である。

Minimax Euclidean Matching of Two Corresponding Sets of Points

Keiko IMAI

Information Science Center, Kyushu Institute of Technology
Iizuka, Fukuoka 820, Japan

We consider the problem of fitting a set S of n points to another similar set T of n points in the plane by rotation and parallel translation, where we suppose that each point of S corresponds to a point of T and this correspondence is one to one. In this paper we present $O(n^2\lambda_{10}(n)\log n)$ algorithm for approximating that minimizes the maximum distance between each point of S and its corresponding point of T by rotation and parallel translation. This problem arises in many practical areas such as pattern recognition, image processing and statistics, etc.

1. はじめに

二つの似かよった点集合の間の最適な当てはめを求める問題は、パターン認識・画像処理・統計などで現われる。画像ロボットによって型 LSI を基板上に自動実装するという例も、このような問題としてとらえられる。この際、決められた場所に部品を機械的に配置するという機械的位置決めに対し、視覚センサを補正機能として用いることが提案されており [7]、実現のためには、センサから得られた幾何的情報をもとにより最適な位置決めをし、実装可能かどうかを判定する必要がある。また、そのようなことが効率よくかつ精密に行えれば、センサを用いて非接触の検査を行なえるという利点を生かすことができる。

視覚機能をもったロボットによる LSI の自動実装問題で、ピングリッドアレイ型 LSI を円形の穴が正方格子状に並んだプリント基板上に自動実装する装置を実現する問題がある ([7]; 図 1 参照)。ここでは、LSI のピンを点とみなした点集合と、円形の穴の中心点の集合との間の対応する点間の距離のミニマックス値が円の半径以下であるとき、かつそのときに限り LSI を基板上に実装することが可能である。

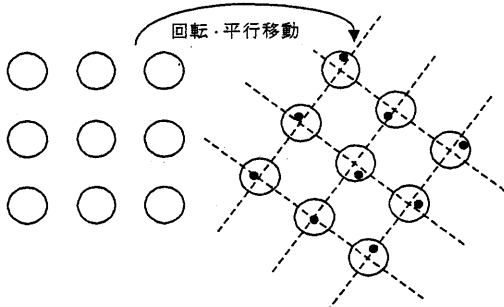


図1. ピングリッドアレイ型 LSI の実装問題

ここでは、この問題を 1 対 1 対応のついた平面上の n 点からなる二つの点集合 S, T が与えられたとき、 S を平行移動と回転することにより対応する点同士の距離の最大値が最小になるようにするミニマックス型の問題としてとらえ、それに対するアルゴリズムを与える。

本稿では、まずこの問題を、 n 個の 3 変数関数の最大値をとる関数の最小化問題として定式化する。 n 個の 1 或は 2 変数関数の最大値 (或は最小値) を値としてとる関数を求めることは、近年の計算幾何学における中心的な話題であり、1 変数の場合は特に Davenport-Schinzel 列という名の下で研究されている [1, 2, 3, 4, 5]。3 変数関数に関する問題は、

まだ殆ど調べられておらず、その点でも以降の議論はおもしろい。この定式化に基づき、ここでの 3 変数関数の最大値関数の性質を調べ、その性質を利用してこの問題に対する $O(n^2 \lambda_{10}(n) \log n)$ の手間のアルゴリズムを与える。ここで、 $\lambda_{10}(n)$ は 10 次の Davenport-Schinzel 列の最大長であり、それ自身は $a(n)$ を Ackermann 関数の逆関数としたとき $O(n 2^{O(a(n)^4)})$ であることが示されている (Davenport-Schinzel 列、Ackermann 関数とその逆関数については付録を参照)。 $\lambda_{10}(n)$ は n に関してほとんど線形に近い関数である。また、以下では三角関数を一部含む定数サイズの多項式の根が定数時間で厳密に求められると仮定する。

関連した問題として、プリント基板上のボタンが円ではなく正方形が格子状に配列された問題があり、その問題に対しては既に $O(n \log n)$ の手間のアルゴリズムを与えられている [6]。

2. 問題の定式化

相異なる n 点よりなる 2 つの点集合

$$S = \{s_i = (x_i, y_i) \mid i = 1, \dots, n\}$$

$$T = \{t_i = (u_i, v_i) \mid i = 1, \dots, n\}$$

において、点 s_i は点 t_i に対応しているとする ($i = 1, \dots, n$)。このとき問題は、 S の点集合を回転移動と平行移動することにより対応する点同士の距離の最大値を最小にすることである。 S の点集合を角度 θ 回転 ($0 \leq \theta < 2\pi$) し、 $(-x, -y)$ 平行移動すると点 s_i の座標は

$$(x_i \cos \theta - y_i \sin \theta - x, x_i \sin \theta + y_i \cos \theta - y)$$

となる。このときの s_i と t_i との距離の二乗を値としてとる関数を $f_i(\theta, x, y)$ とする:

$$f_i(\theta, x, y) = (x - x_i(\theta))^2 + (y - y_i(\theta))^2$$

但し、

$$x_i(\theta) = x_i \cos \theta - y_i \sin \theta - u_i$$

$$y_i(\theta) = x_i \sin \theta + y_i \cos \theta - v_i$$

と定める。このとき、関数 f_i の最大値をとる関数

$$f(\theta, x, y) = \max_i f_i(\theta, x, y)$$

を考えると、このミニマックス型の問題は次のように表される:

$$\min_{\theta, x, y} f(\theta, x, y)$$

$(x_i(\theta), y_i(\theta))$ を座標とする平面上の点を $p_i(\theta)$ とすると、 θ を固定したとき $f(\theta, x, y)$ を x, y に関して最小化する問題は、点 $p_i(\theta)$ ($i = 1, \dots, n$) に対する最小包含円問題である。

3. 関数 $f(\theta, x, y)$ の性質

以下では簡単のため、任意の θ に対して $p_i(\theta)$ ($i=1, \dots, n$) は全て相異なり、全ての θ に対して $p_i(\theta), p_j(\theta), p_k(\theta)$ が一直線上になるような i, j, k の組はないとする。関数 $z=f(\theta, x, y)=\max_i f_i(\theta, x, y)$ を (θ, x, y) 空間に射影すると、 (θ, x, y) 空間の分割が得られる。この分割を n 関数 f_i の最大値図と呼ぶ。この分割は3次元の領域とその2つの交わりで構成される面、3つの交わりで構成される辺、4つの交わりで構成される頂点よりなる。以下では、これらの構成要素の数の上限を与える。

まず、 n 個の3変数関数 $f_i(\theta, x, y)$ の性質を調べる(以下、随時変数を省略して関数を表す)。

(1) 任意の $i \neq j$ に対して $f_i(\theta, x, y)=f_j(\theta, x, y)$ は連続な曲面となる。

(2) 任意の相異なる i, j, k に対して $f_i=f_j=f_k$ は θ をパラメータとする高々2点で不連続な曲線となる((8)参照)。

(3) 任意の相異なる i, j, k, l に対して $f_i=f_j=f_k=f_l$ は高々8個の交点をもつ。

この性質より、最大値図での頂点の数は $O(n^4)$ であることがわかる。よりよい上界は次のようにして得られる。

元の平面での n 点のうち特に $p_i(\theta), p_j(\theta)$ に着目する。 $k \neq i, j$ に対して、関数 $g_k^+(\theta), g_k^-(\theta)$ を次のように定める。ここで、 $p_i(\theta), p_j(\theta), p_k(\theta)$ の外心を q とし、それと線分 $p_i(\theta)p_j(\theta)$ の中点との距離を d とする。

(4.1) $p_k(\theta)$ が有向直線 $p_i(\theta)p_j(\theta)$ の右側にある(図2):

$$g_k^+(\theta) = +d \quad (q \text{ が } p_i(\theta)p_j(\theta) \text{ の右側にある})$$

$$-d \quad (q \text{ が } p_i(\theta)p_j(\theta) \text{ の左側にある})$$

$$g_k^-(\theta) = -\infty$$

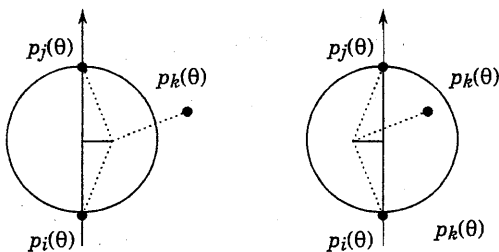


図2. (4.1)

(4.2) $p_k(\theta)$ が有向直線 $p_i(\theta)p_j(\theta)$ の左側にある(図3):

$$g_k^+(\theta) = -\infty$$

$$g_k^-(\theta) = -d \quad (q \text{ が } p_i(\theta)p_j(\theta) \text{ の右側にある})$$

$$+d \quad (q \text{ が } p_i(\theta)p_j(\theta) \text{ の左側にある})$$

(4.3) $p_k(\theta)$ が有向直線 $p_i(\theta)p_j(\theta)$ 上にある(図4):

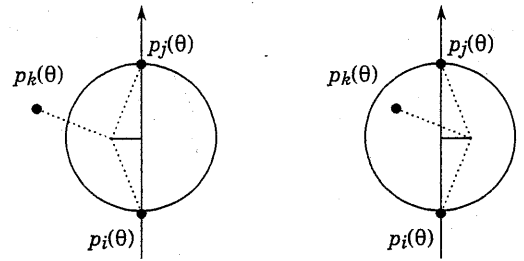


図3. (4.2)

$$g_k^+(\theta)=g_k^-(\theta)=-\infty$$

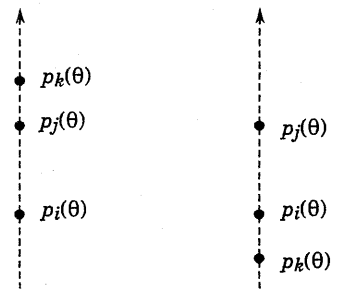


図4. (4.3)

このとき

(5) $\theta=\theta'$ と f_i, f_j, f_k の交点が最大値図上にあるならば、

$$g_k^+(\theta')=\max_{l \neq i, j} g_l^+(\theta')$$

であるか

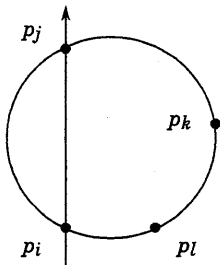
$$g_k^-(\theta')=\max_{l \neq i, j} g_l^-(\theta')$$

である。

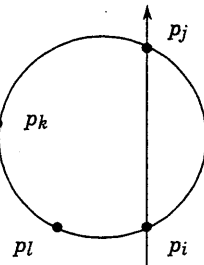
そこで、1変数関数 $g_l^+(\theta)$ ($l \neq i, j$) の最大値関数を g^+ とし、1変数関数 $g_l^-(\theta)$ ($l \neq i, j$) の最大値関数を g^- とする。これらの最大値関数に関して、最大値をとる関数が θ の左右で変わるとき、その θ を区分値という。

f_i, f_j を固定したとき最大値図における $f_i=f_j$ の面上の頂点の数を数えたいのだが、そのかわりに、この区分値の数と $g^++g^-=0$ となる θ の数を数えればよいということが以下の議論でわかる。

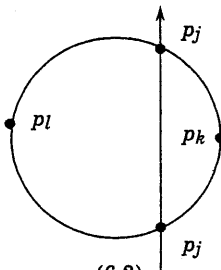
(6) f_i, f_j, f_k, f_l の交点が最大値図上にあり、この交点の θ 座標が θ' であるとする。このとき、 $p_i(\theta'), p_j(\theta'), p_k(\theta'), p_l(\theta')$ は同一円周上にあり他の $p_h(\theta')$ はその円内に含まれている。 $p_i(\theta'), p_j(\theta'), p_k(\theta'), p_l(\theta')$ はその位置関係により次のような3通りが考えられる(図5)。



(6.1)



(6.2)



(6.3)

図5. (6)

(6.1) $p_k(\theta'), p_l(\theta')$ が有向直線 $p_i(\theta')p_j(\theta')$ に関して右側にあるならば、この交点は g^+ の区分値に対応する;

(6.2) $p_k(\theta'), p_l(\theta')$ が有向直線 $p_i(\theta')p_j(\theta')$ に関して左側にあるならば、この交点は g^- の区分値に対応する;

(6.3) $p_k(\theta'), p_l(\theta')$ が有向直線 $p_i(\theta')p_j(\theta')$ に関して別の側にあるならば、この交点は $g^+ + g^- = 0$ の解に対応する;

つまり最大値図の $f_i = f_j$ の面上の点ならば区分値に対応するかまたは $g^+ + g^- = 0$ となる θ に対応している。

逆に、 g^+, g^- の性質から区分値を考えると

(7.1) $g_k^+ = g_l^+ \neq -\infty$ ならば $f_i = f_j = f_k = f_l$ である;

(7.2) $g_k^- = g_l^- \neq -\infty$ ならば $f_i = f_j = f_k = f_l$ である;

また、 $g^+ + g^- = 0$ となっていて g^+, g^- を達成する関数がそれぞれ g_k^+, g_l^- とすると、このときも $f_i = f_j = f_k = f_l$ である。

さらに、

(8) 任意の $k \neq i, j$ に対して、 $p_i(\theta), p_j(\theta), p_k(\theta)$ が同一直線上に存在するような θ は高々 2 つしか存在しない。

すなわち、 g_k^+, g_k^- が $-\infty$ の値をとるところで不連続となるのは高々 2 点である。

以上の議論により、 g^+, g^- での区分値と $g^+ + g^- = 0$ となる θ の数は、 $8 + 2 = 10$ 次の

Davenport-Schinzel 列の最大長のオーダーでおさえられ、最大値図において $f_i = f_j$ の面上にある頂点の数は $O(\lambda_{10}(n))$ であることがいえる。すると、 i, j を相異なる $1, \dots, n$ の整数の対全てについて考えることにより、最大値図での頂点数は、 $O(n^2 \lambda_{10}(n))$ であることがわかる。

定理1. n 関数 f_i の最大値図の頂点数は、 $O(n^2 \lambda_{10}(n))$ である。□

4. 関数 $f(\theta, x, y)$ の最小化

3 節で考察した関数 $f(\theta, x, y)$ の性質を下に、それを最小化することを考える。 θ を固定したとき、 n 点の内 3 点を通る包含円 (全点を含んでいる円) は、3 点を指定されれば唯一であり、全体では通常複数個 (有限) ある。2 点を通る包含円は、通常無限個あるが、2 点を通る包含円としてその 2 点を結ぶ線分を直径とするものしか考えないことにすると、有限個となる。このような 2 或は 3 点を通る有限個の包含円に対して、それを定めている 2 或は 3 個の点の集合を θ に関する包含点集合と呼ぶ。 θ を固定したときの最小包含円を定めている点の集合を、 θ に関する有効包含点集合と呼ぶ。

このとき、

(9) 任意の θ に対して、その有効包含点集合の点数は 2 以上である。

(有効) 包含点集合をその対応する θ の値 (の区間) とともに求めれば、 $f(\theta, x, y)$ を最小化することは容易に行える。以下では、まず 3 節の考察に基づき 2 点を固定した上でそれを含む包含点集合を対応する θ とともに求めるアルゴリズムを与える。また、 θ を 0 から単調に増加させながら対応する有効包含点集合を求めるアルゴリズムも与える。

4.1. 2 점에着目して包含点集合を求めるアルゴリズム

3 節の議論と同様に $p_i(\theta), p_j(\theta)$ に着目し、3 節での記法をそのまま使う。

(10) ある θ に対して、 $g^+(\theta) < 0$ かつ $g^-(\theta) < 0$ であるならば、その θ に対する有効包含点集合は $\{p_i(\theta), p_j(\theta)\}$ である (図 6)。

(11) ある θ に対して、 $g^+(\theta) \geq 0, g^-(\theta) \leq 0$ であり、 g^+, g^- の定義で $\max$ を達成している関数が g_k^+, g_l^- とすると、 $g_k^+(\theta) + g_l^-(\theta) \leq 0$ となっていれば、 $\{p_i(\theta), p_j(\theta), p_k(\theta)\}$ が包含点集合となる (図 7)。逆に、 $g^+(\theta) \leq 0, g^-(\theta) \geq 0$ であり、 $g_k^+(\theta) + g_l^-(\theta) \leq 0$ となっていれば、 $\{p_i(\theta), p_j(\theta), p_l(\theta)\}$ が包含点集合となる。

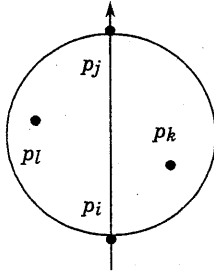


図6. (10)

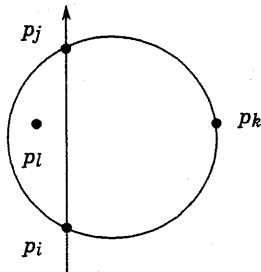


図7. (11)

g^+, g^- の区分値の数は $O(\lambda_{10}(n))$ であった。 g^+, g^- の区分値をマージすると、マージされた区分値の数も $O(\lambda_{10}(n))$ である。さらに (10) の条件 $g^+(\theta) < 0$ かつ $g^-(\theta) < 0$ をみたす区間、(11) の条件 $g^+(\theta) \geq 0$ かつ $g^-(\theta) \leq 0$ であり $g^+(\theta) + g^-(\theta) \leq 0$ となる区間と $g^+(\theta) \leq 0$ かつ $g^-(\theta) \geq 0$ であり $g^+(\theta) + g^-(\theta) \leq 0$ となる区間を求めるためには、 g^+, g^- それぞれと 0 をとる定数関数の交点を求める、符号の判定をすればよい。 g^+, g^- を構成している各極大連結部分と 0 をとる定数関数は n によらない定数回しか交わらないので、全体の交点数は $O(\lambda_{10}(n))$ である。(11) の条件をみたす区間を求めるのは、(10) で求めた $O(\lambda_{10}(n))$ の交点とマージされた区分値と $g^+(\theta) + g^-(\theta) = 0$ の解となる θ によって区切られている区間の内 $g^+(\theta) + g^-(\theta) \leq 0$ となり、かつ (10) の条件をみたさないような区間をみつければよい。

(10) が成り立っている区間では、 $\{p_i(\theta), p_j(\theta)\}$ が有効包含点集合である。(11) が成り立っている区間の θ に対しては、 g^+ または g^- のうちその値が正である関数の方を達成している関数の添字を k とすると、 $\{p_i(\theta), p_j(\theta), p_k(\theta)\}$ が包含点集合となる。

(10), (11) をみたす区間を区切る点の個数については以上でわかったので、それを求めるための手間を評価する。(10), (11) をみたす区間を区切る点を求めるには、 g^+, g^- の区分値と $g^+ + g^- = 0$ が成り立つ θ 、さらに g^+, g^- それぞれと 0 をとる定数関数の交点が求められればよい。まず、このような各点を求める手間について考える。 g^+, g^- の区分値と $g^+ + g^- = 0$ が成り立つ θ を求めるには、 i, j 以外の k, l に対して $f_i = f_j = f_k = f_l$ を解けばよい。三角関数を一部含む定数サイズの多項式の根が定数時間で厳密に求められるという仮定から、 $f_i = f_j = f_k = f_l$ の根は定数時間で求められる。 $g^+ (g^-)$ と 0 をとる定数関数の交点は、 $g^+ (g^-)$ を達成している関数が $g_k^+ (g_l^-)$ とすれば $p_i p_j$ を直径とする円

の円周上に $p_k (p_l)$ があるかどうかを調べればよい。これも定数時間の手間でできる。

次に、(10), (11) の条件をみたす区間全体を求めることを考える。 g^+, g^- を求めるには、 n 関数の最大値をとる関数を求める分割統治法に基づくアルゴリズム (付録 1 参照) を用いればよい。 g^+, g^- を求めるには、元の関数の間の交点は上述のように定数時間ででき、 g^+, g^- の区間の数が $O(\lambda_{10}(n))$ であることから、 $O(\lambda_{10}(n) \log n)$ の手間でできる。先に議論したように、(10) または (11) をみたす区間を区切る点の数は $O(\lambda_{10}(n))$ であり、 g^+, g^- が与えられればマージの手法によりこれらの点を求めることができる。さらに、各区間で (10), (11) をみたすかどうかを判定するのは定数時間でできる。よって、(10), (11) をみたす区間を求め、そのうち (10) をみたす区間では有効包含点集合を、(11) をみたす区間では包含点集合を求めるのは $O(\lambda_{10}(n))$ の手間でできることになる。

以上により、 i, j を固定し、 p_i, p_j が包含点集合に含まれる場合に限った問題のミニマックス問題は、 g^+, g^- を分割統治法で求めるときの手間が $O(\lambda_{10}(n) \log n)$ かかり、他の手順は $O(\lambda_{10}(n))$ で行えるので、全体としては $O(\lambda_{10}(n) \log n)$ の手間でできることになる。

以上の操作を全ての i, j の組について行なえば最小包含円の半径が求められ、考えてきたミニマックス問題が解けたことになる。 n 個から i, j を二つ選んでその i, j に対し $O(\lambda_{10}(n) \log n)$ の手間がかかるので、全体としては $O(n^2 \lambda_{10}(n) \log n)$ にかかることになる。

定理 2. $f(\theta, x, y)$ の最小化問題は、 $O(n^2 \lambda_{10}(n) \log n)$ の手間で解くことができる。□

但し、このアプローチの問題点は、基本的に n 点の問題を n^2 回解くので、 $\Omega(n^3)$ の手間がどうしてもかかってしまうことである。

4.2. θ を単調に増加させながら対応する有効包含点集合を求めるアルゴリズム

4.1 節のアルゴリズムよりは、当面最悪の場合の手間は悪いが、平均的にはより高速であろうと思われる次のようなアルゴリズムが考えられる。 θ を 0 から 2π まで動かしたとき、有効点集合は θ の幾つかの値のところで不連続に変わる。そのような θ の値を、臨界値と呼ぶ。一般に、適当な値 θ に対して、 θ を次第に増加させていったとき初めてあたる臨界値は、 $O(n)$ の手間で求めることができる。また、4.1 節の議論を適用することにより、臨界値の総数は $O(n^2 \lambda_{10}(n))$ であることがいえる。すると、 $\theta=0$ から始めて、臨界値を小さい順に求め

ていきながら $f(\theta, x, y)$ を最小化するアルゴリズムが考えられる。

このアルゴリズムの手間は、上の議論からだけでは $O(n^3 \lambda_{10}(n))$ であることしかいえないが、臨界値の総数は上の上界よりかなり小さいと予想されるので、実際的には効率よいと考えられる。

5. まとめ

本稿では、 n 個の 3 変数関数の最大値関数の性質を調べるというアプローチにより、対応の与えられた点集合間のミニマックス近似問題がある程度効率よく解けることを示した。但し、最大値関数に関して示した上界等はきつものであるかどうかはわかっておらず、改良の余地が残っている。また、ここでは簡単のため、任意の θ に対して $p_i(\theta)$ ($i=1, \dots, n$) は全て相異なるとしたが、実際はある θ に対していくつかの $p_i(\theta)$ が一致することが起こりうる。そのような場合についての議論も今後必要である。

さらに、本稿では点集合 S を平行移動と回転によって点集合 T に当てはめることを考えたが、平行移動、回転に拡大縮小も加えて当てはめを求める問題も考えられる。

$$x_i(\theta, r) = x_i \cos \theta - y_i \sin \theta - u_i r$$

$$y_i(\theta, r) = x_i \sin \theta + y_i \cos \theta - v_i r$$

とおき

$$f_i(\theta, x, y, r) = (x - x_i(\theta, r))^2 + (y - y_i(\theta, r))^2$$

を考え、 f_i の最大値をとる関数

$$f(\theta, x, y, r) = \max_i f_i(\theta, x, y, r)$$

を最小にする問題:

$$\min_{\theta, x, y, r} f(\theta, x, y, r)$$

を考える。この場合には、 n 個の 4 変数関数の最大値関数を最小化するという問題を扱うことになり興味深い。

付録1. n 関数の最小値をとる関数 — Davenport-Schinzel 列

n 個の 1 変数連続関数 $f_i(x)$ の集合で、任意の対の関数は互いに高々 s 個の点で交わるものを考える。この関数の集合に対して、それらの関数の最小値を値としてとる関数を f とする:

$$f(x) = \min_i f_i(x)$$

関数 f のグラフは、 n 関数 f_i のグラフの下側エンベロープ (lower envelope) であり、この f のグラフは、エンベロープ上の関数 f_i のグラフの極大連結

部分の合併として表わせる。このときの極大連結部分の数の最大値を $\lambda_s(n)$ で表わす。

エンベロープ上で隣接する異なる関数の極大連結部分は、それらの関数の交点である一つの点を共有し、また関数 f_i は互いに高々 s 回しか交わらないという仮定より、

$$\lambda_s(n) \leq sn(n-1)/2 + 1$$

であり、 $\lambda_s(n)$ は常に定義できる。しかし、 s を n に関係のない定数とみなしたとき、この上界は n の二乗に比例するものであり、上界としてかなり緩いものであることが予想される。実際、 $\lambda_s(n)$ に関する次のような上限、上界が与えられている (例えば、[4, 5])。

$$\lambda_1(n) = n$$

$$\lambda_2(n) = 2n - 1$$

$$\lambda_3(n) = \Theta(n \alpha(n))$$

$$\lambda_4(n) = \Theta(n 2^{\alpha(n)})$$

$$\lambda_{2s}(n) = O(n 2^O(\alpha(n)^{s-1})), s > 2$$

$$\lambda_{2s+1}(n) = O(n \alpha(n) O(\alpha(n)^{s-1})), s \geq 2$$

$$\lambda_{2s}(n) = \Omega(n 2^{\Omega(\alpha(n)^{s-1})}), s > 2$$

ここで、 $\alpha(n)$ は Ackermann 関数の逆関数で、非常に増加するのが遅い関数である (付録 2 参照)。 s を n に関係のない定数として、簡潔に書くと

$$\lambda_s(n) = O(n \log^* n)$$

である。 $\log^* n$ も $\log n$, $\log \log n$ などよりもずっと増加するのが遅い関数である (付録 2 参照)。

例を示す。 n 関数 f_i が、全て d 次の多項式であるとき、任意の二つの関数は高々 d 回しか交わらないので、その下側エンベロープは $\lambda_d(n)$ 個の各関数のグラフの極大連結部分からなっている。 $d=1$ のとき、 f_i は線形であり、下側エンベロープを求めることは、 f_i のグラフの直線が上より定める半平面の交わりを求める問題に対応する (図 8)。また、ロ

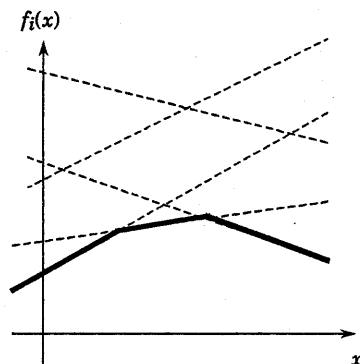


図8. 線形関数の下側エンベロープ

ボティックスなどへの応用では、 n 直線分の下側エ

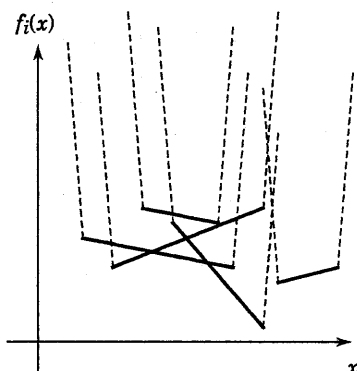


図9. n 線分の下側エンベロップ

ンベロップを求める問題としてよく現われる(図9)。各線分は、それぞれ異なる区間上でのみ定義された線形関数とみなせる。上の議論では、各関数の定義域は同じで、すべて連続関数でなければいけなかった。そこで、線分に対応する関数を次のようにして実数全体で定義された区分的線形関数とする。グラフ上で、各線分の左端点から $-\infty$ に近い同じ傾きで半直線を延ばし、右端点から $+\infty$ に近い同じ傾きで半直線を延ばす(図9参照)。得られた n 連続区分的線形関数の下側エンベロップは、元の n 線分の下側エンベロップと位相的に等価であり、前者を求めることにより後者を構成することができる。これらの区分的線形関数は互いに高々3回しか交わらず、 n 線分の下側エンベロップが $O(n\alpha(n))$ 個の部分よりなることもわかる。

与えられた n 関数 f_i に対して、それらのグラフの下側エンベロップを求めるアルゴリズムとしては、分割統治法(divide-and-conquer method)を適用したものが知られている。それによれば、 $O(\lambda_s(n) \log n)$ の手間で下側エンベロップの各関数の極大連結部分を左から右にソートした順に求めることができる。本文中の g^+ を例にアルゴリズムを簡単に書くと次のようになる。

1. $(n-2)$ 個の θ の関数 $g_k^+(\theta)$ の集合をほぼ等しい数の二つの集合に分ける;
2. 分割した二つの集合に対し再帰的にアルゴリズムを適用し、 θ の区分値によって区切られた各区間で最大値を与える関数から構成される関数 $\tilde{g}^+, \hat{g}^+$ を求める;
3. $\tilde{g}^+$ と $\hat{g}^+$ の最大値をとる関数 g^+ を求める;

ステップ3において、 $\tilde{g}^+$ と $\hat{g}^+$ をマージして g^+ を求めるには、区間の数に比例する手間がかかる。従って、このアルゴリズムは全体で $O(\lambda_{10}(n))$

$\log \lambda_{10}(n))$ つまり $O(\lambda_{10}(n) \log n)$ の手間でできる。

$\lambda_s(n)$ を評価する際には、等価な次のような組合せ問題として議論されている。正の整数 n, s に対して、次の三つの条件を満たす整数列 $U=(u_1, u_2, \dots, u_m)$ を、 (n, s) Davenport-Schinzel 列という:

- (i) $1 \leq u_i \leq n$ ($i=1, \dots, m$)
- (ii) $u_i \neq u_{i+1}$ ($i=1, \dots, m-1$)
- (iii) $u_{i_1}=u_{i_3}=u_{i_5}=\dots=a$, $u_{i_2}=u_{i_4}=u_{i_6}=\dots=b$, $a \neq b$ であるような $s+2$ 個の添字 $1 \leq i_1 < i_2 < \dots < i_{s+2} \leq m$ が存在しない。

すると、 $\lambda_s(n)$ は、 (n, s) Davenport-Schinzel 列の長さの最大値に等しい。

以上の議論は全て1変数関数に関するものであり、その研究も上で述べたようにほぼ一段落している。現在では、多変数の問題が精力的に調べられており、いくつかの結果が出ている。本文中では元の問題の性質より n 個の3変数関数の最大値をとる関数を求めることを考え、それに関連して最大値図を定義した。同様に、 n 個の d 変数関数の集合 F に対して、 F の関数の最小値をとる関数を求める問題においては、 n 関数の d 次元での最小値図を定義することができる。そのとき、最小値図でのファセットの数 $\kappa(F)$ を評価すること、最小値図を構成することが問題となる。

多変数関数に関する結果をあまり厳密にはないが簡単にまとめると、次のようになる。 n 個の2変数関数 $f_i(x, y)$ で、任意の三つの関数が高々 s 点で交わる場合については、

- (i) 任意の二つの関数の交わりの曲線が、 x をパラメタとする連結な単純曲線であり、任意の三つの関数は高々1点で交わるならば、 $\kappa(F)=O(n)$ で、最小値図を $O(n \log n)$ の手間で求められる。
- (ii) 任意の二つの関数の交わりが連結な単純曲線であり、任意の三つの関数は高々2点で交わるならば、 $\kappa(F)=O(n^2)$ 。
- (iii) 任意の二つの関数の交わりの曲線が、 $x=\text{定数}$ と表わせる平面と高々2点で交わるならば、 $\kappa(F)=O(n\lambda_{s+2}(n))$ 。

一般の次元に関しては、

- (iv) $d+1$ 次元での n 個の d 次元単体の下側エンベロップでの構成要素の数は、 $O(n^d \alpha(n))$ である($d=1$ の場合が、2次元での n 線分の下側エンベロップを求める問題)。

n 関数の最小値をとる関数を求める問題は、本文中で示した応用の他に、既にロボットの障害物回避経路の探索問題など幅広い問題に応用されている。

