アルゴリズム 9－4
(1989. 7. 27)

多次元 Davenport-Schinzel 列計算における線形化手法とその応用

今井桂子
九州工業大学・情報科学センター

1次元Davenport-Schinzel列計算の問題は，n 個の1変数関数の最小値をとる関数を求める問題として，最近の計算幾何学の中心的話題の１つである．しかし，多次元 Davenport-Schinzel 列，及び n 個の多変数関数の最小値をとる関数を求める問題については，研究が始まったばかりである．本稿では，多次元 Davenport-Schinzel 列計算における線形化手法を紹介し，n 個の多変関数の最小値をとる関数を求めるアルゴリズムを与える．また，多次元 Davenport-Schinzel 列計算における線形化手法の平面上を回転と平行移動で動く点に対する最小包含円問題や平面上の対応の与えられたつの点集合を回転と平行移動によって最適な位置に当てはめる問題等への応用についても考察し，これらの問題を $O(n^3 \log n)$ の手間，$O(n^3)$ の記憶領域で解くアルゴリズムを与える．

On the Linearization Technique in Computing Multidimensional Davenport-Schinzel Sequences and Its Applications

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

We consider the problem of computing multidimensional Davenport- Schinzel sequences by using the linearization technique. This problem has strong connection with the combinatorial complexity of the lower (or upper) envelope of n multivariate functions. We have well known about the one-dimensional Davenport-Schinzel sequences (or, the envelope of single variable functions). But, the combinatorial complexity of the lower (or upper) envelope of multivariate functions has not yet been analyzed except a few cases [5]. In this paper, we present the linearization technique and its applications to the dynamic enclosing circle problem of moving points in the plane and the minimax geometric fitting problem of two corresponding sets of points.

1. はじめに

n 個の 1 変数あるいは 2 変数関数の最小値（あるいは最大値）をとる関数を求めることは，近年の計算幾何学の中心的話題となっている．特に，1 変数の場合は Davenport-Schinzel 列という名の下で研究されている [1,2,3,4,5,6,13]．3 変数以上の関数に関する問題は，研究が始まったばかりである．本稿では，n 個の多変数関数の最小値（あるいは最大値）をとる関数を求める問題，つまり，多次元 Davenport-Schinzel 列計算についての線形化手法を紹介し，その線形化手法を用いて Attallah によって紹介された動的計算幾何学 (dynamic computational geometry) の 1 つの問題である平面上を動いている点に対して，それらの点の最小包含円を求める問題（点が回転と平行移動により動く場合）や，パタン認識・画像処理・統計などでよく現われる，2 つの平面上の対応が与えられている点集合を一方の集合を回転と平行移動によりもう一方の集合に最適な位置に当てはめる問題 (geometric fitting problem) を解くアルゴリズムを与える．

2. n 個の 1 変数関数の最小値をとる関数と Davenport-Schinzel 列

まず，1 変数関数の最小値をとる関数について知られている結果についてまとめておく．

n 個の 1 変数連続関数 $f_i(x)$ の集合で，任意の二つ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:

$$f(x) = \min_{i=1,\ldots,n} f_i(x)$$

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から構成されている．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 1 つの点を共有している．このことと，関数 f_i は互いに高々 s 回しか交わらないという仮定から，

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

であることはすぐわかるが，より良い上限，上界が得られている (例えば，[1,5,6,13])．

$$\begin{aligned}
&\lambda_1(n) = n \\
&\lambda_2(n) = 2n - 1 \\
&\lambda_3(n) = \Theta(n\alpha(n)) \\
&\lambda_4(n) = \Theta(n2^{\alpha(n)}) \\
&\lambda_{2s}(n) = O(n2^{O(\alpha(n)^{s-1})}),\ s > 2 \\
&\lambda_{2s+1}(n) = O(n\alpha(n)^{O(\alpha(n)^{s-1})}),\ s \ge 2 \\
&\lambda_{2s}(n) = \Omega(n2^{\Omega(\alpha(n)^{s-1})}),\ s > 2
\end{aligned}$$

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．s を n に関係ない定数として，簡潔に表わすと

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

となる．$\log^* n$ は，n に対して 2 を底として対数を繰り返しとったとき初めて 1 以下になるときの回数を表わす関数だが，これも $\log n$, $\log\log n$ などよりもずっと増加するのが遅い関数である．従って，$\lambda_s(n)$ はほとんど線形に近い関数であるといえる．

例えば，n 関数 f_i が，すべて d 次多項式であるとき，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 $\lambda_s(n)$ 個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 の直線の定める上側の半空間の交わりを求める問題に対応する．

また，ロボティックスなどへの応用では，n 直線分の下側エンベロープを求める問題がよく現われる．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考えられる．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ならなかった．そこで，各線分の左端点から $-\infty$ に近い同じ傾きで半直線をのばし，右端点からも $+\infty$ に近い同じ傾きで半直線をのばして，線分に対応する関数を実数全体で定義された区分的線形関数に拡張する．このようにして得られた n 個の連続な区分的線形関数の下側エンベロープを求めれば，元の n 線分の下側エンベロープが構成できる．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ープが $\lambda_3(n) = (n\alpha(n))$ 個の部分からなることもわかる．

与えられた n 個の関数 f_i に対して，それらのグラフの下側エンベロー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．この方法を用いれば $O(\lambda_s(n)\log n)$ の手間で下側エンベロープの極大連結部分を左から右へソートした順に求めることができる．また，不連続な点がある場合についても，各関数が互いに高々 s 回しか交わらないときは，$O(\lambda_{s+1}(n)\log n)$ の手間で求められる ([7])．

$\lambda_s(n)$ 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3つの条件を満たす整数列$U = (u_1, u_2, \ldots, u_m)$を，$(n, s)$Davenport-Schinzel 列という：

$$\text{(i) } 1 \le u_i \le n (i = 1, \ldots, m)$$

$$\text{(ii) } u_i \ne u_{i+1} (i = 1, \ldots, m-1)$$

$$\begin{aligned}\text{(iii) } & u_{i_1} = u_{i_3} = u_{i_5} = \ldots = a, \\ & u_{i_2} = u_{i_4} = u_{i_6} = \ldots = b, a \ne b\end{aligned}$$

であるような $s+2$ 個の添字 $1 \le i_1 < i_2 < \ldots < i_{s+2} \le m$ が存在しない．すると，$\lambda_s(n)$ は，(n, s) Davenport-Schinzel 列の長さの最大長に等しい．

3. n 個の多変数関数の最小値をとる関数と線形化手法

1 節で見たように n 個の 1 変数関数の最小値をとる関数についての研究は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，いくつかの結果が得られてはいるが，それらは，扱っている関数の固有の性質に依存するもので，1変数の場合のように一般的な結果は，まだあまり知られていない．n 個の多変数関数の最小値をとる関数を構成する問題，つまり，多次元 Davenport-Schinzel 列を計算する問題についての研究は，始まったばかりといえる．

n 個の d 変数関数 f_i の最小値をとる関数

$$f(x_1, \ldots, x_d) = \min_{i=1,\ldots,n} f_i(x_1, \ldots, x_d)$$

の$d+1$次元空間内でのグラフ$z=f(x_1,\ldots,x_d)$をd次元空間内に射影すると，$(x_1,\ldots,x_d)$空間の分割が得られる．この分割をn個の関数の集合$F=\{f_1(x_1,\ldots,x_d),\ldots,f_n(x_1,\ldots,x_d)\}$の最小値図と呼ぶ．$F$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$d$次元での最小値図の構成要素（ファセット，facet）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になる．

n個の関数f_iは、次のような性質を満たすとする．

1) 任意のk個の関数のグラフの交わりは$d-k+1$次元となる．

2) すべてのf_iに共通な成分を除けば，各f_iは，

$$a_{i0}+\sum_{k=1}^{d}a_{ik}X_k$$

と書き換えられる．但し，a_{ik}は定数であり，X_kは，変数$x_1,\ldots,x_d$とその三角関数からなるiには依らない多項式であり，各X_kの関係はすべて，有限個の多項式で与えられるとする．

このように関数f_iをX_kの線形関数で表わすと，n個のd変数関数f_iの最小値をとる関数

$$f(x_1,\ldots,x_d)=\min_{i=1,\ldots,n}f_i(x_1,\ldots,x_d)$$

を求める問題は，

$$\min_{i=1,\ldots,n}a_{i0}+\sum_{k=1}^{m}a_{ik}X_k$$

という，X_kの関係式から成る非線形な制約式を持つ線形関数の最小化問題に等価となる．

1)，2)の性質を満たす関数f_iの例として，次のようなものを考える．

$$\begin{aligned}f_i(\theta,x,y)=&x^2+y^2+x_i^2+y_i^2+u_i^2+v_i^2+2u_ix+2v_iy\\&+2(-u_ix_i-v_iy_i)\cos\theta+2(u_iy_i-v_ix_i)\sin\theta\\&+2x_i(-x\cos\theta-y\sin\theta)+2y_i(x\sin\theta-y\cos\theta)\end{aligned}$$

とすれば，x^2+y^2はすべての関数に共通なので取り除き，残りの部分は，X_k,a_{ik}を

$$X_1=x,\quad X_2=y,\quad X_3=\cos\theta,\quad X_4=\sin\theta$$
$$X_5=-x\cos\theta-y\sin\theta,\quad X_6=x\sin\theta-y\cos\theta.$$

および

$$a_{i0}=x_i^2+y_i^2+u_i^2+v_i^2,\qquad a_{i1}=2u_i,\qquad a_{i2}=2v_i$$
$$a_{i3}=2(-u_ix_i-v_iy_i),\qquad a_{i4}=2(u_iy_i-v_ix_i),\qquad a_{i5}=2x_i,\qquad a_{i6}=2y_i$$

と置くことにより，

$$a_{i0}+\sum_{k=1}^{6}a_{ik}X_k$$

と書くことができる．$X_1,\ldots,X_3$の間には，

$$X_3^2+X_4^2=1$$
$$X_5=-X_1X_3-X_2X_4$$
$$X_6=X_1X_4-X_2X_3$$

という関係がある．つまり，

$$\min_{i=1,\ldots,n} f_i(\theta, x, y)$$

を求める問題は，次のようなつの非線形な制約式を持つ線形関数の最小化問題に等価である．

$$\begin{aligned}
\min_{i=1,\ldots,n} \ & a_{i0} + \sum_{k=1}^{6} a_{ik} X_k \\
\text{s.t.} \ & X_3^2 + X_4^2 = 1 \\
& X_5 = -X_1X_3 - X_2X_4 \\
& X_6 = X_1X_4 - X_2X_3
\end{aligned}$$

この問題を例にとって線形手法について説明していくことにする．

まず，7 次元 Euclid 空間内で n 個の線形関数 $Z = a_{i0} + \sum_{k=1}^{6} a_{ik}X_k$ の下側エンベロープを考え，このエンベロープと 3 個の制約式の交わりが丁度最小値図と対応している．

一般に，m 次元 Euclid 空間内の n 個の半空間の交わりの境界を構成する要素の数は $O(n^{\lfloor m/2 \rfloor})$ であることから，今考えている下側エンベロープのサイズは $O(n^3)$ であることがわかる．下側エンベロープは，7 次元凸多面体 (convex polyhedron) であり，フェイシャル・グラフ (facial graph) が定義できる．フェイシャル・グラフとは，1 つの入口 (source) と 1 つの出口 (sink) をもつアサイクリックグラフ (acyclic graph) であり，その各点 (node) は下側エンベロープのフェイス (face) を表わし，あるフェイスが別のフェイスの構成要素になっているとき，そのフェイスどうしを枝で結んだものである．このようにして定義されるフェイシャル・グラフは，Seidel[12] の凸包のアルゴリズムによって構成することができる．最悪の場合には，フェイシャル・グラフを構成するのには，$O(n^3 \log n)$ の手間がかかり，その大きさは，$O(n^3)$ となる．

今，下側エンベロープと 3 つの制約式との交わりを直接求めるのではなく，フェイシャル・グラフを使って，最小値図を下側エンベロープと 3 つの制約式を用いて構成して行く方法をとる．最小値図上の頂点は、n 個のうち 4 個の関数の交点であるから，これを 7 次元 Euclid 空間で考えれば，4 つの超平面の交わりからなる 3 次元フェイスと 3 つの制約式の交点となる．同様に，最小値図の辺，面は，それぞれ，下側エンベロープの d 次元フェイス（$d = 4,5$）と 3 つの制約式の交わりと対応している．

まず，最小値図の頂点を計算する．対応する 3 次元フェイスは，n 個の 7 次元内の超平面のうちある 4 個の超平面の交わりから成る 3 次元空間を，残りの $n-4$ 個の超平面によって定められる半空間で制限したものであった．この 3 次元フェイスと 3 つの制約式との交わりが最小値図上の頂点と対応しているので，任意の 4 個の超平面をとり，その共通部分である 3 次元空間と 3 つの制約式とのすべての交点（候補点，candidate point）をまず計算する．候補点はすべてが下側エンベロープ上にあるとは限らないので，それが下側エンベロープ上にあるかどうかを判定しなければならない．3 次元フェイスの数は $O(n^3)$ あるので，各 3 次元フェイスに対し残りの $n-4$ 個の超平面のどちら側にあるかを調べると $O(n^4)$ の手間がかかってしまう．そこで，この判定にフェイシャル・グラフを用いる．

3 次元フェイスの構成要素となっている 2 次元フェイスは，関数の性質 1) より，3 次元フェイスを構成している 4 個の超平面ともう 1 つ，かつただ 1 つの超平面の交わりで定義

されている．このような今考えている3次元フェイスの構成要素である2次元フェイスを定義している超平面についてのみ，候補点のチェックを行なえばよい．一般に，m 次元内の凸多面体の k 次元フェイスの数と，k 次元フェイスと $k+1$ 次元フェイスの間の枝の数は，どちらも $O(n^{\min\{\lfloor m/2\rfloor, k+1\}})$ である．よって，フェイシャル・グラフが与えられていれば，このチェックは定数時間でできる．

最小値図上の枝と面を求めるには，下側エンベロープの d 次元フェイス $(d=4,5)$ と3つの制約式との交わりを求めればよい。これも、フェイシャル・グラフを用いて $O(n^3 \log n)$ の手間で求めることができる。

また，既に述べたように，Seidel のアルゴリズムにより，フェイシャル・グラフは $O(n^3 \log n)$ の手間で求められる．そして，その大きさは最悪の場合 $O(n^3)$ である．よって，この例では，線形化手法を用いると $O(n^3)$ の記憶領域，$O(n^3 \log n)$ の手間で n 個の3変数関数の最小値をとる関数を求めることができる．

4．ミニマックス問題への応用

4.1 動的な点に対する最小包含円問題

平面上に与えられた n 個の点の最小包含円を求める問題は，計算幾何学の基本的問題の1つであり，これは，$O(n)$ の手間，$O(n)$ の記憶領域で解けることが知られている [9]．また，点が異なる直線上を一定の速度で動いている場合の最小包含円を求めるのも，$O(n(\log n)^4 \log\log n)$ の手間，$O(n)$ の記憶領域でできる [10]．このような動的な点に対して問題を考えることが，ロボティックス，グラフィックスなどへの応用を扱うために必要になってきている．

ここでは，平面上に与えられた n 個の点が θ 回転した後，平行移動される場合の最小包含円を求める場合を考える．3節で述べた線形化手法により，以下で示すように点が回転と平行移動により動く場合の最小包含円問題は $O(n^3 \log n)$ の手間，$O(n^3)$ の記憶領域で解くことができる．

平面上の n 個の点 $p_i = (x_i, y_i)$ $(i = 1, \ldots, n)$ が与えられているとする．これらの点 $p_i = (x_i, y_i)$ が θ 回転 $(0 \leq \theta < 2\pi)$ された後，(u_i, v_i) 平行移動されるとすると，回転，平行移動後の点 p_i 座標は

$$\begin{aligned} p_i(\theta) =&(x_i(\theta), y_i(\theta)) \\ &x_i(\theta) = x_i \cos\theta - y_i \sin\theta + u_i \\ &y_i(\theta) = x_i \sin\theta + y_i \cos\theta + v_i \end{aligned}$$

となる．点 $p = (x, y)$ と $p_i(t)$ の Euclid 距離の二乗をとる関数

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

を考える．θ を θ_0 に固定すると，n 個の3変数関数 $f_i(\theta_0, x, y)$ の最大値をとる関数

$$f(\theta_0, x, y) = \max_{i=1,\ldots,n} f_i(\theta_0, x, y)$$

が最小になる点 $p = (x, y)$ が $\theta = \theta_0$ における最小包含円の中心である．さらに，θ を動かして最小値を求めれば，点が回転と平行移動によって動いている場合の最小包含円問題が解けたこ

とになる．つまり，この n 個の 3 変数関数 $f_i(\theta,x,y)$ の最大値をとる関数の最小値を求めるという問題 (以下，TR ー最小包含円問題という)

$$\min_{0\leq\theta<2\pi,\, x,\, y}\ \max_{i=1,\ldots,n} f_i(\theta,x,y)$$

を解くのと同等である．

点 $p=(x,y)$ と $p_i(\theta)$ の Euclid 距離の二乗をとる関数

$$\begin{aligned} f_i(\theta,x,y) =&(x-x_i(\theta))^2+(y-y_i(\theta))^2\\ =&x^2+y^2+x_i^2+y_i^2+u_i^2+v_i^2+2u_ix+2v_iy\\ &+2(-u_ix_i-v_iy_i)\cos\theta+2(u_iy_i-v_ix_i)\sin\theta\\ &+2x_i(-x\cos\theta-y\sin\theta)+2y_i(x\sin\theta-y\cos\theta)\end{aligned}$$

は 3 節で例として扱った関数に他ならない．従って，3 節では最小値図を考えたが，そのかわりに最大値図を考えれば，3 節で紹介した線形化手法を用いて n 個の関数の最大値をとる関数は $O(n^3\log n)$ の手間，$O(n^3)$ の記憶領域で求められる．そして，その最小値を計算すれば求める解が得られる．

4.2 2 つの対応の与えられた点集合間のミニマックス近似問題

2つの似かよった点集合の最適の当てはめを求める問題(geometric fitting problem)は，パターン認識・画像処理・統計といった分野でよく現われる．例えば，画像ロボットによってピングリッドアレイ型 LSI を基盤上に自動実装するという問題が一例としてあげられる．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盤に自動実装する装置を実現する問題がある [11]．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において，対応する点の間の距離の最大値が最小になるように基盤を実装する問題である．

このような問題において，一方の集合を回転と平行移動によってもう一方の集合に当てはめる問題を考えると，丁度，本稿で扱った TR- 最小包含円問題に帰着することができる．

5. おわりに

本稿では，n 個の関数の最大値をとる関数の最小値をとる問題を線形化手法を用いて解く方法について紹介した．3 節で例にとって説明した問題を別のアルゴリズムを用いて（1 変数の Davenport-Schinzel 列の問題に帰着して解く方法）解くと，問題を解く手間は線形化手法より少し劣るが（$O(n^2\lambda_7(n)\log n)$ 時間），$O(n)$ の記憶領域で解くことができる [8]．従って，4 節で応用としてあげた TR- 最小包含円問題や対応の与えられた 2 つの点集合の最適当てはめ問題も，こちらの方法でも解くことができる．

TR- 最小包含円問題では，回転と平行移動で点が動く場合について考えたが，回転され，その後，一定の速度で直線上を動く，つまり，点 p_i の座標が

$$\begin{aligned} p_i(\theta,\alpha) =&(x_i(\theta,\alpha),y_i(\theta,\alpha))\\ &x_i(\theta,\alpha)=x_i\cos\theta-y_i\sin\theta+\alpha u_i\\ &y_i(\theta,\alpha)=x_i\sin\theta+y_i\cos\theta+\alpha v_i\end{aligned}$$

で表わされる場合 (TSR- 最小包含円問題) にも，線形化手法を使うことができる．この問題は，$O(n^4 \log n)$ の手間，$O(n^4)$ の記憶領域で解ける．TSR- 最小包含円問題は，1 変数の Davenport-Schinzel 列の問題に帰着して解くよりも，線形化手法の方がいまのところ効率が良いという結果がでている [8]．もちろん，TSR- 最小包含円問題に対しても，対応する点集合の当てはめ問題が存在し，TSR- 最小包含円問題と同じ手法を用いて解くことができる．

謝辞

本研究の一部は、文部省科学研究費奨励研究 (A) の援助を受けた。

参考文献

[1] P. Agarwal, M. Sharir and P.Shor: Sharp upper and lower bounds on the length of general Davenport-Schinzel sequences. Manuscript, 1988.

[2] M. J. Atallah: Some dynamic computational geometry problems. *Computers and Mathematics with Applications*, Vol.11, No.12 (1985), pp.1171–1181.

[3] H. Davenport and A. Schinzel: A combinatorial problem connected with differential equations. *American Journal of Mathematics*, Vol.87 (1965), pp.684-694.

[4] H. Edelsbrunner: The upper envelope of piecewise linear functions: Tight bounds on the number of faces. *Technical Report No.UIUCDCS-R-87-1396*, Department of Computer Science, University of Illinois at Urbana-Chanpaign, December 1987.

[5] H. Edelsbrunner, J. Pach, J.T. Schwartz and M. Sharir: On the lower envelope of bivariate functions and its applications. *Proceedings of the 28th IEEE Annual Symposium on Foundations of Computer Science*, 1987, pp.27–37.

[6] S. Hart and M. Sharir: Nonlinearity of Davenport-Schinzel sequences and of generalized path compression schemes. *Combinatorica*, Vol.6 (1986), pp.151–177.

[7] J. Hershberger: Finding the upper envelope of n line segments in $O(n \log n)$ time. Manuscript, 1988.

[8] K. Imai, S. Sumino and H. Imai: Minimax geometric fitting corresponding sets of points. *Proceedings of the fifth Annual Symposium on Computational Geometry*, 1989, pp.266-275.

[9] N. Megiddo: Linear programming in linear time when the dimension fixed. *Journal of the Association for Computing Machinery*, Vol.31(1984), pp.114-127.

[10] N. Megiddo: Dynamic location problems. *Annals of Operations Research*, Vol.6 (1986), pp.313-319.

[11] S. Naito: Applications of visual sensors to industrial robots (in Japanese). *Sensor Technology*, Vol.7(1987), pp.90-93.

[12] R. Seidel: Output-size sensitive algorithms for constructive problems in computational geometry. Ph. D. Thesis, Department of Computer Science, Cornell University, 1986.

[13] E. Szemeredi: On a problem by Davenport and Schinzel. *Acta Arithmetica*, Vol.25 (1974), pp.213-224.