有限点集合の凸包を求める効率のよいアルゴリズム

（卒業論文）

（指導教官　小島 政和 助教授）　　東京工業大学理学部情報科学科　田村　明久

1. はじめに

コンピュータ・グラフィックス，自動設計やパターン認識などに応用されるものとして有限点集合の凸包を求める問題が知られているが，本論文では有限点集合の凸多面体のすべてのファセットを求める問題をとり扱う．

ここでファセットとは，d次元凸多面体と支持超平面の交わりで定義されるフェイスのうち次元が$d-1$のものであり，フェイスの次元はフェイスのアフィン包の次元で定義される．また，これは数理計画法における不等式系から端点集合を求める問題の双対な問題であり，ファセットは不等式の形で表現できる．

この問題については，1970年にChand-Kapurの提案したアルゴリズム(以後CK法と略す)があるが，CK法は単純な方法であるため無駄が多い．本論文では，CK法を基本とし点集合を辞書式順序に並べかえて2つの方法によって効率化を試みる．1つは，端点を順々に固定していくことと，内点の検出を処理中に行なうことにより効率化をはかる．もう1つは，点集合を次々に2分割して小問題化していくことにより効率をよくする．

2. CK法

CK法では，「d次元凸多面体の$(d-2)$次元フェイスは，ちょうど2つのファセットに含まれる」という性質を使い，初期ファセットから不飽和な$(d-2)$次元フェイス（これを含むファセットが1つしか求まっていないもの）を次々にたどることによりすべてのファセットを求めている．入力点集合S，次元$d(\dim S=d)$に対しCK法　CK(S,d)　は次のようになる．

$S=\{P^i \mid P^i=(p_1{}^i,\cdots,p_d{}^i)\in R^d,$
$\quad i\neq j\Leftrightarrow P^i\neq P^j(i,j=1,\cdots,n)\}$

CK(S,d)｛

　初期ファセットfとこれに含まれる$(d-2)$次元フェイスの集合Cを求める；

$E\leftarrow C$；　$F\leftarrow\{f\}$；

while $E\neq\phi$｛

　Eのある$(d-2)$次元フェイスとこれを含むファセットfから$S-f$の点を対象にして新しいファセットを求める；

　ファセットf^*に含まれる$(d-2)$次元フェイスの集合Cを求める；

　$E\leftarrow E\triangle C$；　$F\leftarrow F\cup\{f^*\}$；

｝　　[△：対称差]

｝

ファセットの更新においては，$(d-1)\times d$連立方程式をいちど解き，入力点数nに対し$2\times(n-d)$回程度内積を計算する必要があるが，以下ではおもにファセットの更新において対象となる点数を減らすことにより効率化をはかる．

3. 辞書式順序を導入したアルゴリズム（LO法と略）

ここでは，入力点集合Sが辞書式順序に並んでいるとする．すなわち，

$S=\{P^i \mid P^i=(p_1{}^i,\cdots,p_d{}^i)\in R^d,$
$\quad i<j\Leftrightarrow P^i\lesssim P^j(i,j=1,\cdots,n)\}$
$\quad$ただし，$\dim S=d$，$\lesssim$：辞書式順序

改良されたところは，以下の2点である．

(1) P^1から順々に端点を固定し，固定された端点を含むファセットをすべて求め，以後の比較ではこの端点を対象外とする．

(2) (1)の処理中にP^iがすでに求まっているどのファセットの端点にもならないとき，P^iはSの凸包の端点にもならない(自明ではない)ので以後の比較ではP^iを対象外とする．

Sが辞書式順序に並んでいる利点は，

$P^i\lesssim P^j\Leftrightarrow h(P^i)<h(P^j)(i,j=1,\cdots,n)$ となるような線形関数$h(x)$が存在することである．この性質は(2)や次のアルゴリズムに利用されている．

CK法では内点は処理が終わるまでわからない．特に

入力点数に対し端点数が少ないときに(2)は有効である.

4. 点集合の分割を利用したアルゴリズム(SD法と略)

ここでも,入力点集合Sが辞書式順序に並んでいるとする.ここでは,点集合を2分割して小問題化することにより効率化をはかる.以下にSD法を簡単に示す.

(1) Sを$S_1=\{P^1,\cdots,P^m\}$, $S_2=\{P^{m+1},\cdots,P^n\}$に分割する.$m=\llcorner n/2\lrcorner$

(2) Sの凸包のファセットで,S_1の点もS_2の点も端点としてもつものをすべて求める.

(3) Sの凸包のファセットで,S_1の点のみを端点としてもつものをすべて求める.

(4) Sの凸包のファセットで,S_2の点のみを端点としてもつものをすべて求める.

ただし,(3),(4)は再帰的に実行する.

SD法では,(2)の処理が速さを決定する.すなわち,(2)で求めるファセットの数が少ないほど効率がよいが,どのような点集合の分割が理想的なファセットの分割を与えるかわからない.それゆえ,SD法は速さのばらつきがあることが予想できる.また点集合が辞書式順序に並んでいることは,ファセットの集合をそれぞれ(2),(3),(4)で求めるものに分割できることを保証する.

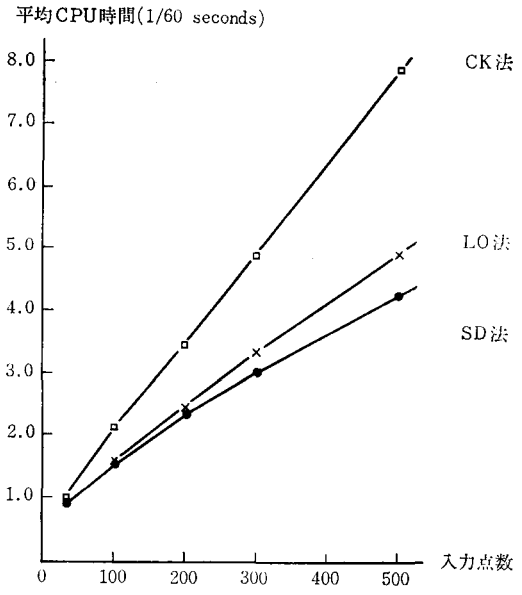
5. 実験結果

図は,CK法,LO法,SD法の効率を比較するため4次元超球面上にランダムに発生させたデータの数に対する1ファセット当りの平均CPU時間を示す.
(VAX11/780, Unix, C言語による)

他の実験結果からも計算量のオーダーは同じだがLO法,SD法ともに点数がある程度以上になると,CK法より速いことがわかる.LO法の特徴は,データに対するばらつきがCK法と同様に小さいことがあげられる.それに対し,SD法はデータに対する1ファセット当りの平均内積回数のばらつきが大きい欠点はあるが,不飽和な$(d-2)$次元フェイスの集合の更新時のフェイスとフェイスの比較回数はCK法,LO法に比べて少ない.

6. さいごに

実際の応用においては,フェイス束を必要とするときもあるが,このような場合にも$(d-3)$次元以下のフェイスは組合せ的に求めることができる.入力データによっては,LO法がSD法より速い場合がある.端点の選び方を工夫することによりSD法にLO法の改良と同様のことを導入することも可能と思われる.



参考文献

[1] A.V.Aho, J.E.Hopcroft and J.D.Ullman: アルゴリズムの設計と解析I,(野崎昭弘,野下浩平(共訳),サイエンス社)

[2] D.R.Chand and S.S.Kapur:"An algorithm for convex polytopes" J. ACM, 17, 7 (Jan. 1970), 78-86

[3] K.Fukuda: "An Efficient Pivot Algorithm for Finding All Edges and All Vertices of a Convex Polytope: 3-Dimensional Case", Res. Rep. Inf. Sci., TIT, B-131 (1983)

[4] B. Grunbaum: Convex Polytopes, Wiley, New, York, 1967

[5] P. McMullen and G. C. Shephard: Convex Polytope and the Upperbound Conjecture, Cambridge University Press, 1971

[6] F. P. Preparata and S. J. Hong: "Convex hulls of finite sets of points in two and three dimensions", Comm. ACM., 20 (1977), 87-93