アルゴリズム 22－4
（1991. 7. 22）

ソートされた点集合の凸包を求める最適並列アルゴリズム

陳慰 中野浩嗣 増澤利光 都倉信樹
大阪大学基礎工学部情報工学科

本稿では，まず，平面上のソートされた点集合の凸包を求める問題について考察し，以下の並列アルゴリズムを示す．(1) EREW PRAM 上の $n/\log n$ プロセッサ，$O(\log n)$ 時間のアルゴリズム．(2) CRCW PRAM 上の $n/\log\log n$ プロセッサ，$O(\log\log n)$ 時間のアルゴリズム．(3) CRCW PRAM 上の $n^{1+\epsilon}$ プロセッサ，$O(1)$ 時間のアルゴリズム．アルゴリズム (1) と (2) は最適であり，アルゴリズム (3) は，プロセッサ数を多く用いるが，より高速である．また，アルゴリズム (1),(2),(3) に簡単な変更を加えることにより，同じ計算量で，単純多角形の凸包を求めるアルゴリズムが得られる．

Optimal Parallel Algorithms for Finding the Convex Hull of a Sorted Point Set

Wei Chen, Koji Nakano,Toshimitsu Masuzawa, and Nobuki Tokura
Faculty of Engineering Science,Osaka University,Osaka,560 Japan

This paper presents several parellel algorithms for finding the convex hull of a sorted planar point set: (1) the algorithm which runs in $O(\log n)$ time using $n/\log n$ processors on a EREW PRAM, (2) the algorithm which runs in $O(\log\log n)$ time using $n/\log\log n$ processors on a CRCW PRAM, (3) the algorithm which runs in $O(1)$time using $n^{1+\epsilon}$ processors on a CRCW PRAM. The algorithms (1) and (2) are optimal,and if enough processors are available, the algorithm (3) is faster. Moreover,without increasing the complexity, algorithms (1),(2)and (3) can be simply modified to find the convex hull of a simple polygon, respectively.

1 まえがき

Sを2次元平面上のn個の点の集合とするとき，Sの凸包$CH(S)$ (*convex hull*) とは，Sを包含する最小の凸多角形である．Sの与えられ方の違いによっていくつかの凸包問題が考えられる．

(一般凸包問題) Sが任意のn個の点の点集合で与えられたとき，$CH(S)$を求める問題．

(ソート点凸包問題) Sがソートされたn個の点（例えば，x座標でソートされる）の点集合で与えられたとき，$CH(S)$を求める問題．

(多角形凸包問題) 単純n角形Pが与えられたとき，Pのn個の頂点からなる集合Sに対して，$CH(S)$を求める問題．

凸包問題は計算幾何学における基本的な問題としてよく研究されている．ソート問題が一般凸包問題に帰着できるので，一般凸包問題の計算時間の下界は$\Omega(n \log n)$である．ソート点凸包問題と多角形凸包問題の計算時間の自明な下界は$\Omega(n)$である．これらの凸包問題それぞれについて，時間計算量が下界と一致する逐次アルゴリズムが知られている[10]．

並列計算モデル，特に，PRAM(Parallel Random Acccess Machine) の上で，凸包問題を解くアルゴリズムの研究も活発に行なわれている[1],[3],[4],[5],[7],[9],[12]．本稿では，PRAM の上で，ソート点凸包問題と多角形凸包問題の最適並列アルゴリズムを示す．PRAM とは並列計算モデルであり，複数のプロセッサが一つの共有メモリに接続している．各プロセッサは，共有メモリの任意のメモリセルにアクセスできる．このとき，一つのメモリに対して複数のプロセッサが同時に読みだしを行なったり，書き込みを行なったりすることが考えられる．これら同時読みだしや同時書き込みの制限によって，3 つの PRAM モデル，CRCW（同時読みだし可，同時書き込み可．Concurrent Read Concurrent Write），CREW（同時読みだし可，同時書き込み不可．Concurrent Read Exclusive Write），EREW（同時読みだし不可，同時書き込み不可．Exclusive Read Exclusive Write）が提案されている．CRCW モデルは，同時書き込みの制約によって，さらにいくつかのモデルに細分されるが，本稿では Common モデル（同じ値のときのみ，同時書き込みが許される）のみを扱う．PRAM の 3 つのモデルでは，CRCW は最も能力が高く，EREW は最も能力が低いモデルである．つまり，モデル M を用いて，コスト（プロセッサ数と計算時間の積）cで，入力サイズnのある問題を解く時間を$T^M(n,c)$とするとき，$T^{EREW}(n,c) \geq T^{CREW}(n,c) \geq T^{CRCW}(n,c)$が成り立つ．また，一つのモデル上のアルゴリズムは，より能力の高いモデルの上で，同じプロセッサ数と計算時間で実行できる．本稿ではこの性質をよく利用する．並列アルゴリズムのコストが，その問題を解く最適逐次アルゴリズムの時間計算量とオーダ的に一致するとき，その並列アルゴリズムのコストが最適であるという．さらに，ある並列アルゴリズムが最適とはコストが最適であるという条件を満たした上で時間計算量が最小であるという意味である．

一般凸包問題を CREW,EREW それぞれで，n プロセッサ，$O(\log n)$ 時間で解く最適並列アルゴリズムがいくつか知られている[1],[4],[5],[12]．一般凸包問題を CRCW で，多項式プロセッサ，$O(1)$ 時間[2]，$(n-d)C_d^n$プロセッサ，$O(d)$ 時間[13]で解く並列アルゴリズムも知られている．ソート点凸包問題と多角形凸包問題については，次の結果が知られている．

(a) ソート点凸包問題が，(i) CREW で，$n/\log n$ プロセッサ，$O(\log n)$ 時間[9]，(ii) CRCW で，$n \log\log n/\log n$ プロセッサ，$O(\log n/\log\log n)$ 時間で解ける[7]．

(b) 多角形凸包問題が，CREW で，$n/logn$ プロセッサ，$O(\log n)$ 時間で解ける[3]．

一般に，凸包問題の結果$CH(S)$の出力形式として次の2つが考えられる．1つは$CH(S)$（凸多角形）の頂点に印を付ける形式で，識別形式 (Identification) と呼ぶ．もう1つは$CH(S)$の頂点を時計回りの順に列挙する形式で，構造形式 (Construction) と呼ぶ．構造形式は，さらに，$CH(S)$を配列で表す形式（配列構造形式と呼ぶ）とリストで表す形式（リスト構造形式と呼ぶ）に分けられる．前述の3つの最適逐次アルゴリズムの時間計算量のオーダは，結果の出力形式に依存しない[10]．しかし，並列アルゴリズムでは，出力の形式によって計算量の異なることがある．同じ凸包問題に関して，明らかに，$CH(S)$の識別形式は最も求めやすく，$CH(S)$の配列構造形式を求めるのは最も難しい．表1にソート点凸包問題と多角形凸包問題の各出力形式ごとの計算量をまとめておく．

表1 凸包問題に関する並列アルゴリズムの計算量
(太い線で囲まれている結果は本研究の結果)

(a) 最適並列アルゴリズムの時間計算量
(プロセッサ数=n/時間計算量)

問題	出力の記述	時間計算量 CRCW	CREW	EREW
ソート点凸包問題	配列構成形式	$\Theta(\log n/\log\log n)$ (6)	$\Theta(\log n)$ (8)	$\Theta(\log n)$
	表記とリスト構成形式	$\Theta(\log\log n)$	$\Theta(\log n)$ (8)	$\Theta(\log n)$
多角形凸包問題	配列構成形式	$\Theta(\log n/\log\log n)$	$\Theta(\log n)$ (8)	$\Theta(\log n)$
	表記とリスト構成形式	$\Theta(\log\log n)$	$\Theta(\log n)$ (8)	$\Theta(\log n)$

(b) CRCW上の定数時間の並列アルゴリズム

問題	出力の記述	プロセッサ数
ソート点凸包問題	表記とリスト構成形式	$n^{1+\epsilon}$
多角形凸包問題	表記とリスト構成形式	$n^{1+\epsilon}$

本稿では，まず，ソート点凸包問題の解をリスト構造形式で求める以下の並列アルゴリズムを提案する．

(1) EREW 上の$n/\log n$プロセッサ，$O(\log n)$時間のアルゴリズム EREW-SP．

(2) CRCW 上の$n/\log\log n$プロセッサ，$O(\log\log n)$時間のアルゴリズム CRCW-SP．

(3) CRCW 上の$n^{1+\epsilon}$（$\epsilon > 0$は任意の正の実定数）プロセッサ，$O(1)$時間のアルゴリズム CRCW-SPCT．

アルゴリズム EREW-SP，CRCW-SP それぞれは，EREW と CRCW 上の最適アルゴリズムである．アルゴリズム CRCW-SPCT は CRCW 上で定数時間でソート点凸包問題を解くので，プロセッサ数が多く使えるような状況では，他のアルゴリズムより高速である．また，表 1(a) に示したように，アルゴリズム EREW-SP，CRCW-SP は出力形式を識別形式にしたときも最適である．さらに，接頭部和 (*prefixsum*) を計算することにより，リストを配列に格納することが，EREW で$O(n/\log n)$プロセッサ，$O(\log n)$[6]時間で，CRCW で$O(n \log\log n/\log n)$プロセッサ，$O(\log n/\log\log n)$[6]時間でできる．従って，アルゴリズム EREW-SP，CRCW-SP は出力形式を配列構造形式にしたときも最適である．文献 (7) のアルゴリズムでは，出力形式をリスト構造形式にすると，簡単には，最適並列アルゴリズムにならない．本稿で提案するアルゴリズムでは，基本的には，文献 (7) のアルゴリズムのアイデアを用いるが，最適並列アルゴリズムにするため，より効率的な新しいデータ構造を採用する．

本稿では，次に，多角形凸包問題のリスト構造形式を求める並列アルゴリズムを提案する．単純n角形が時計回りの順の頂点の系列で与えられたとき，ある程度ソートされた平面点集合と見なせる．従って，ソート点凸包問題に対するアルゴリズムに簡単な変更を加えることにより，多角形凸包問題に対しても(1),(2),(3)と同様に，以下の結果が得られる．

(4) EREW 上の $n/\log n$ プロセッサ，$O(\log n)$ 時間のアルゴリズム EREW-POL.

(5) CRCW 上の $n/\log\log n$ プロセッサ，$O(\log\log n)$ 時間のアルゴリズム CRCW-POL.

(6) CRCW 上の $n^{1+\epsilon}$ プロセッサ，$O(1)$ 時間のアルゴリズム CRCW-POLCT.

本稿は次のように構成される．本稿で提案するアルゴリズムでは，2つの凸多角形の共通接線を求めることが重要な役割を果たす．そこで，2章では，まず，配列に格納された凸多角形の共通接線を求めるアルゴリズムを示す．3章では，ソート点凸包問題を解くアルゴリズムを示す．4章では，ソート点凸包問題のアルゴリズムを多角形凸包問題のアルゴリズムに変更する方法を示す．5章では，アルゴリズムの最適性を示す．

2 凸多角形の共通接線を求めるアルゴリズム

$U=u_0,u_1,\ldots,u_{h_1-1}$ と $V=v_0,v_1,\ldots,v_{h_2-1}$ を2つの交わらない凸多角形とし，これらの頂点それぞれが時計回りの順に配列に格納されているとする．文献(5)は，U と V の共通接線を，CREW で，$h^{1/c}$ (但し，$h=max(h_1,h_2)$) プロセッサ，$O(c^2)$ 時間で求めるアルゴリズムを提案している．文献(5)のアルゴリズムを CREW の上で f プロセッサで実行すれば，時間計算量が $O((\log h/\log f)^2+1)$ になる．この章では，まず，文献(5)のアルゴリズムを示す．次に，このアルゴリズムのいくつかの拡張について述べる．

一般性を失うことはなく，U と V が1本の垂直線で左と右に分けられると仮定する．一般に，U と V の共通接線は4つあるが，ここでは，上側共通接線（U と V がともに下にある共通接線）を求める方法のみを示す．他の共通接線も同様に求められる．以下では，上側共通接線を単に共通接線と呼ぶ．凸多角形 U と V の共通接線の U 上の接点を u^*，V 上の接点を v^* とする．u^* と v^* を求めるために，次の性質を利用する．

性質 1 (5) 凸多角形 U' と V' それぞれを $U'=u_{i_1},u_{i_2},\ldots,u_{i_s}(0\le i_1<i_2<\cdots<i_s<h_1-1)$，$V'=v_{j_1},v_{j_2},\ldots,v_{j_t}(0\le j_1<j_2<\cdots<j_s<h_2-1)$，$U'$ と V' の共通接線の U' 上の接点を $u_{i_{k_1}}$，V' 上の接点を $v_{j_{k_2}}$ とする．このとき，凸多角形 U'' を $U''=u_{i_{k_1-1}+1},u_{i_{k_1-1}+},\ldots,u_{i_{k_1+1}}$ [1]，凸多角形 V'' を $V''=v_{j_{k_2-1}+1},v_{j_{k_2-1}+2},\ldots,v_{j_{k_2+1}}$ とすれば，次のいずれか（すくなくとも一方）が成り立つ．

(1) u^* が U'' の頂点

(2) v^* が V'' の頂点 □

性質 2 (5) 凸多角形 U' を $U'=u_{i_1},u_{i_2},\ldots,u_{i_s}$，$U'$ と V の共通接線の U' 上の接点を u_{i_k} とする．そのとき，u^* が $U''=u_{i_{k-1}+1},u_{i_{k-1}+2},\ldots,u_{i_{k+1}}$ にある． □

まず，これらの性質を用いて CREW PRAM で f プロセッサで U と V の共通接線を求めるアルゴリズム CREW-CTAN(f) を述べる．

{アルゴリズム CREW-CTAN(f)}

[入力] 配列に格納された凸多角形 $U=u_0,u_1,\ldots,u_{h_1-1}$ と $V=v_0,v_1,\ldots,v_{h_2-1}$.

[出力] 凸多角形 U と V の共通接線の U 上の接点 u^* と V 上の接点 v^*.

[方法] (ステップ0) $\hat{U}:=U,\hat{V}:=V,\hat{h_1}:=h_1,\hat{h_2}:=h_2$.

(ステップ1) $max(\hat{h_1}^2,\hat{h_2}^2)\le f$ のとき，次のように $\hat{U}$ と $\hat{V}$ の共通接線を求める．U と V の頂点からなる各頂点対 (u_i,v_j)（このような頂点対は全体で h_1h_2 個（高々 f 個）ある）に1つずつプロセッサを割り当て，各直線 u_iv_j（u_i と v_j を通る直線）が $\hat{U}$ と $\hat{V}$ の共通接線であるか否かを1つのプロセッサを用いて調べる．直線 u_iv_j が U の接線であるための必要十分条件は，$\hat{U}$ の辺 $u_{i-1}u_i$ と u_iu_{i+1} が直線 u_iv_j の下にあることである（$\hat{V}$ についても同様）．$max(\hat{h_1}^2,\hat{h_2}^2)>f$ のとき，以下のステップを実行する．

(ステップ2) (1) $k_1=max(\hat{h_1}/f^{1/2},1)$, $k_2=max(\hat{h_2}/f^{1/2},1)$ とする．U と V からそれぞれ，k_1,k_2 ごとに頂点を取って得られた凸多角形を U', V' とする．つまり，$U'=u_0,u_{k_1},u_{2k_1},\ldots$, $V'=v_0,v_{k_2},v_{2k_2},\ldots$ とする．ステップ1と同じ方法を用いて U' と V' の共通接線を求める．得られた共通接線を直線 $u_{ik_1}v_{jk_2}$ とする（図1）．

(2) 凸多角形 U'' を $U''=u_{(i-1)k_1+1},u_{(i-1)k_1+2},\ldots,u_{(i+1)k_1}$ とする．U と V の共通接線の U 上の接点が U'' にあるかどうかを次のように判定する．

手続き TAN（後述）を用いて，$u_{(i-1)k_1+1}$ から $\hat{V}$ に引いた接線 T_1，$u_{(i+1)k_1}$ から V に引いた接線 T_2 を求める．T_1 が U の接線であれば，$u_{(i-1)k_1+1}=u^*$ である．T_2 が U の接線であれば，$u_{(i+1)k_1}=u^*$ である．T_1,T_2 が U の接線でなければ，U 上の接点 u^* が U'' にあるための必要十分条件は，$u_{(i-1)k_1}$ が T_1 の下にあり，かつ，$u_{(i+1)k_1+1}$ が T_2 の上にあることである．

U と V の共通接線の U 上の接点が U'' にあるならば，U を U'', h_1 を $|U''|$ とする．同様に，凸多角形 $V''=v_{(j-1)k_2+1},v_{(j-1)k_2+2},\ldots,v_{(j+1)k_2}$ に対して，U と V の共通接線の V 上の接点が V'' にあるかどうかを判定し，あるならば，V を V'', h_2 を $|V''|$ とする（性質1より，U と V の共通接線の接点は，U'', V'' の少なくとも一方にある）．

(ステップ3) ステップ1，2を用いて再帰的に U と V の共通接線を求める． □

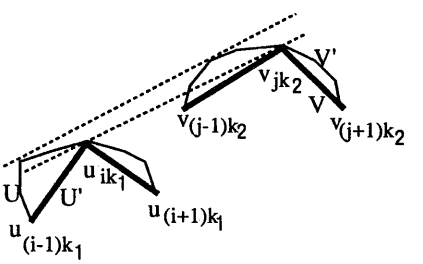

図1. 2つの凸経路の共通接線

次に，凸多角形 $W=w_0,w_1,\ldots,w_{h_3}$ と w の外の点 p が与えられたとき，p から W に引いた接線を f プロセッサ，$O(\log h_3/\log f+1)$ 時間で求める手続き TAN を示す．

{手続き TAN}

(ステップ1) $h_3\le f$ のとき，W の各頂点 w_i に1つのプロセッサを割り当てる．直線 pw_i が W の接線であるかどうかを判定することより，p から W に引いた接線を f プロセッサ，$O(1)$ 時間で求める．

(ステップ2) $h_3>f$ のとき，W から h_3/f ごとに頂点を取って得られた凸多角形を W' とする．つまり，$k=h_3/f$ とすると，$W'=w_0,w_k,w_{2k},\ldots$ である．ステップ1の方法を用いて，p から W' に引いた接線を求める．この接線の接点を w_{ik} とするとき，性質2より，p から W に引いた接線の接点は $W''=w_{(i-1)k+1},w_{(i-1)k+2},\ldots,w_{(i+1)k}$ にある．そこで，凸多角形 W'' に対して再帰的にこの

[1] 本稿では，任意の系列 $X=(x_1,x_2,\ldots,x_k)$ に対して，混乱が生じないときに，$x_{(a+b)mod k}$ を x_{a+b}，$x_{(a-b)mod k}$ を x_{a-b} $(1\le a,b\le k)$ と略記する．

アルゴリズムを適用し，p から W に引いた接線を求める． □

手続き TAN のステップ2では，再帰的に手続き TAN を実行するが，このとき，凸多角形の頂点数が $2h_3/f$ になる．従って，手続き TAN の計算時間を $T(h_3)$ とすると，

$$T(h_3) = \begin{cases} T(2h_3/f) + O(1), & h_3 > f \\ O(1), & h_3 \leq f \end{cases}$$

が成り立つ．よって，手続き TAN の計算時間は，$T(h_3) = O(\log h_3/\log f + 1)$ である．

次に，アルゴリズム CREW-CTAN(f) の計算時間を解析する．ステップ1は $O(1)$ 時間で実現できる．$h = max(h_1, h_2)$ とすると，ステップ2の一回の実行には手続き TAN より $O(\log h/\log f + 1)$ 時間かかる．性質2より，ステップ2を一回実行した後，U，或は，V のサイズは元のサイズの $1/f^{1/2}$ になる．よって，ステップ2は $O(\log h/\log f)$ 回実行される．従って，アルゴリズム CREW-CTAN(f) の計算時間は $O((\log h/\log f)^2 + 1)$ である．

定理 1 配列で記述されている 2 つの凸 h 角形の共通接線は，*CREW* の上で f プロセッサ，$O((\log h/\log f)^2 + 1)$ 時間で求められる． □

3 個以上の凸多角形が与えられたときに，各凸多角形対に f プロセッサを割り当てれば，アルゴリズム CREW-CTAN(f) を利用して，各凸多角形対の共通接線が求められる（このアルゴリズムを CREW-ACTAN(f) と呼ぶ）．よって，次の系が成り立つ．

系 1 x 個の凸 h 角形 U_i $(1 \leq i \leq x)$ が配列で与えられたとき，全ての凸多角形対 (U_i, U_j) $(1 \leq i < j \leq x)$ の共通接線は，*CREW* の上で $x^2 f$ プロセッサ，$O((\log h/\log f)^2 + 1)$ 時間で求められる． □

次に，アルゴリズム CREW-CTAN(f) を EREW で実行できるように，アルゴリズム EREW-CTAN(f) に変更する．

{**アルゴリズム EREW-CTAN(f)**}

[入力と出力] アルゴリズム CREW-CTAN(f) と同じ．

[方法] アルゴリズム CREW-CTAN(f) ではステップ1，2でのみ同時読みだしが起こる．ただし，ステップ2で同時読みだしが起こるのは，ステップ1の方法を用いて共通接線を求める場合であり，ここではステップ1についてのみ考える．ステップ1を次のように変更すれば，CREW-CTAN(f) は EREW の上で実行できる．

アルゴリズム CREW-CTAN(f) のステップ1では，各 (u_i, v_j) $(0 \leq i \leq h_1 - 1, 0 \leq j \leq h_2 - 1)$ に1つのプロセッサ（P_{ij} とする）を割り当て，u_i, v_j が U と V の共通接線であるか否かを調べる．このとき，プロセッサ $P_{i1}, P_{i2}, \ldots, P_{i(h_2-1)}$ が同時に u_i を，$P_{1j}, P_{2j}, \ldots, P_{(h_1-1)j}$ が同時に v_j を読み出す．この同時読みだしを避けるために，(a) $P_{i1}, P_{i2}, \ldots, P_{i(h_2-1)}$ の間に値 u_i を，(b) $P_{1j}, P_{2j}, \ldots, P_{(h_1-1)j}$ の間に値 v_j を同時読みだしと同時書き込みがないように転送する．(a) は $P_{i1}, P_{i2}, \ldots, P_{i(h_2-1)}$ の上で二分バランス木を作れば，$O(\log f)$ 時間でできる．(b) についても同様である． □

アルゴリズム EREW-CTAN(f) の計算時間の解析はアルゴリズム CREW-CTAN(f) と同様である．ただし，ステップ1とステップ2を一回実行するための時間計算量は $O(\log f + \log h/\log f)$ になる．よって，アルゴリズム EREW-CTAN(f) の時間計算量は $O(\log^2 h/\log^2 f + \log f + \log h)$ になる．

定理 2 配列に格納された 2 つの凸 h 角形の共通接線は，*EREW* で，f プロセッサ，$O(\log^2 h/\log^2 f + \log f + \log h)$ 時間で求められる． □

次に，x 個の凸 h 角形 $U_i = u_0^i, u_1^i, \ldots, u_{h-1}^i$ $(1 \leq i \leq x)$ が配列で与えられたとき，全ての凸多角形対 (U_i, U_j) $(1 \leq i < j \leq x)$ の共通接線を EREW の上で求めることを考える．CREW-ACTAN(f) では，各凸多角形対に f 個のプロセッサを割当て，独立にアルゴリズム CREW-CTAN(f) を実行した．しかし，複数の凸多角形対を同時に処理しているので，各凸多角形対に f 個のプロセッサを割当て，独立にアルゴリズム EREW-CTAN(f) を実行すると同時読みだしが生じる場合がある．例えば，凸多角形対 (U_i, U_j) の共通接線を求めるプロセッサと $(U_i, U_{j'})$ の共通接線を求めるプロセッサが，U_i の同じ頂点に対して同時読みだしを行なうことがある．以下では，各凸多角形対の共通接線を EREW で求めるために，これらの同時読みだしを避ける処理を加えたアルゴリズム EREW-ACTAN(f) を示す．アルゴリズム EREW-ACTAN(f) では，凸多角形対 (U_i, U_j) の共通接線を求める途中段階では U_i 及び U_j の連続した部分頂点系列（簡単のため，以下では，連続した部分頂点系列を単に連続した一部と呼ぶ）のみを扱う．この連続した一部を表すために，U_i, U_j それぞれに連続した一部の最初と最後の頂点を表す2つのポインタ $b(i,j), e(i,j)$ と $b(j,i), e(j,i)$ を用いる．つまり，このとき，U_i の連続した一部を $U(i,j)$，U_j 連続した一部を $U(j,i)$ とすると，$U(i,j) = u_{b(i,j)}^i, u_{b(i,j)+1}^i, \cdots, u_{e(i,j)}^i$，$U(j,i) = u_{b(j,i)}^j, u_{b(j,i)+1}^j, \ldots, u_{e(j,i)}^j$ である．

{**アルゴリズム EREW-ACTAN(f)**}

[入力] 配列に格納された凸 h 角形 $U_i = u_0^i, u_1^i, \ldots, u_{h-1}^i$ $(1 \leq i \leq x)$．

[出力] 全ての凸多角形対 U_i, U_j $(1 \leq i < j \leq x)$ の共通接線．

[方法] 各 i, j $(1 \leq i, j \leq x$ かつ $i \neq j)$ に対して，j に関する U_i の連続した一部を $U(i,j) = u_{b(i,j)}^i, u_{b(i,j)+1}^i, \cdots, u_{e(i,j)}^i$，$i$ に関する U_j の連続した一部を $U(j,i) = u_{b(j,i)}^j, u_{b(j,i)+1}^j, \ldots, u_{e(j,i)}^j$ とする．全ての凸多角形対 $(U(i,j), U(j,i))$ の共通接線を同時に求める方法を示す．全ての凸多角形 (U_i, U_j) $(1 \leq i < j \leq x)$ の共通接線は，$b(i,j) := 0, e(i,j) := h - 1$ に初期設定することによって求められる．

（ステップ1） (1) 各 i,j $(1 \leq i < j \leq x)$ に対して，$h(i,j) := e(i,j) - b(i,j) + 1$ とする．$h(i,j) \neq 0$ かつ $max(h(i,j)^2, h(j,i)^2) \leqq f$ のとき，EREW-CTAN(f) のステップ1の方法で，f プロセッサ，定数時間で $U(i,j)$ と $U(j,i)$ の共通接線を求める．このとき，複数の凸多角形対を同時処理するために生じる同時読みだしを次のように解消する．

方法としては，アルゴリズム EREW-CTAN(f) と同様に，同じ値を読むプロセッサの上で二分バランス木を作ってこの値を伝送する．ただし，アルゴリズム EREW-CTAN(f) では，同じ値を読むプロセッサが連続している（つまり，プロセッサを表す変数 P の添字が連続している）ので，簡単に二分木を構成できたが，EREW-ACTAN(f) では，連続していない．従って，ここでは，まず，同じ値を読むプロセッサを次のように連続させる．各 i,j $(1 \leq i, j \leq x$, かつ，$i \neq j)$ に対して，$U(i,j)$ を $u_{b(i,j)}^i, u_{b(i,j)+1}^i, \cdots, u_{e(i,j)}^i$ とする．ステップ1では，共通接線を定数時間で求めるため，$U(i,j)$ の各要素 u_r^i $(b(i,j) \leq r \leq e(i,j))$ に一つのプロセッサ $P_{i,r,j}$ を割り当てる．同じ要素に割り当てられるプロセッサを連続させるため，3次組の集合 $ID = \{(i,r,j) | 1 \leq i, j \leq x$, かつ，$i \neq j$, かつ，$b(i,j) \leq r \leq e(i,j)\}$ を辞書順序によってソートする（$x^2 f$ プロセッサ，$O(logx + \log f)$ 時間でできる[8]）．このとき，同じ i,r を持つ組 $(i,r,j_1), (i,r,j_2), \ldots,$ が連続する．よって，同じ頂点 $u_{b(i,j)+r}^i$ に割り当てるプロセッサ $P_{i,r,j_*}, Pi,r,j_2, \ldots,$ が連続する．次に，同じ値を読むプロセッサの上で二分バランス木を作って，木の上でこの値を伝送する（$x^2 f$ プロセッサ，$O(\log x + \log f)$ 時間でできる）．

(2) $h(i,j) := 0, h(j,i) := 0$ にする．

(3) 任意の i,j $(1 \leq i < j \leq x)$ について，$h(i,j) = 0$ であれば，アルゴリズムを終了する．

（ステップ2） 各 i,j $(1 \leq i < j \leq x)$ に対して，$max(h(i,j)^2, h(i,j)^2) > f$ のとき，

(1) $k(i,j) = max(h(i,j)/f^{1/2}, 1), k(j,i) = max(h(j,i)/f^{1/2}, 1)$ とし，$U(i,j)$ と $U(j,i)$ それぞれから $k(i,j), k(j,i)$ ごとに頂点を

取って得られた凸多角形を$U'(i,j)$,$U'(j,i)$とする．$U'(i,j)$と$U'(j,i)$の共通接線をステップ1 (1)の方法で求める．この共通接線を$(u^i_{ak(i,j)}, u^j_{bk(j,i)})$とする．

(2) $U(i,j)$の$u^i_{(a-1)k(i,j)+1}$と$u^i_{(a+1)k(i,j)}$の間の頂点からなる凸多角形を$U''(i,j)$とする．つまり，$U''(i,j)$=$u^i_{(a-1)k(i,j)+1}$, $u^i_{(a-1)k(i,j)+2}$,…, $u^i_{(a+1)k(i,j)}$である．$U(i,j)$と$U(j,i)$の共通接線の$U(i,j)$上の接点が$U''(i,j)$にあるかどうかを手続きTANを用いて判定する（複数個の凸多角形対に対して同時に手続きTANを実行するため，同時読みだしが生じる場合がある．このときの処理はステップ1 (1)と同様）．$U(i,j)$と$U(j,i)$の共通接線の$U(i,j)$上の接点が$U''(i,j)$にあるならば，$b(i,j) := (a-1)k(i,j)+1$,$e(i,j) := (a+1)k(i,j)$,$h(i,j) := 2k(i,j)$にする（このことは，凸多角形$U(i,j)$の更新を意味する）．$U(j,i)$についても同様に処理する．

（ステップ3）再帰的に全ての凸多角形対$U(i,j)$と$U(j,i)$（$1 \le i < j \le x$）の共通接線を求める． □

アルゴリズムEREW-ACTAN(f)の解析を行なう．時間計算量の解析はアルゴリズムCREW-CTAN(f)と同様に行なえる．ただし，ステップ1とステップ2を一回実行するための時間計算量が$O(\log x + \log f)$になる．よって，アルゴリズムEREW-ACTAN(f)の時間計算量は$O((\log^2 h/\log^2 f + 1)(\log x + \log f))$になる．アルゴリズムEREW-ACTAN(f)で用いるプロセッサ数は明らかに$x^2 f$である．

定理 3 x個の凸h角形U_i（$1 \le i \le x$）が配列で与えられたとき，全ての凸多角形対(U_i, U_j)（$1 \le i < j \le x$）の共通接線は，*EREW PRAM*の上で$x^2 f$プロセッサ，$O((\log^2 h/\log^2 f+1)(\log x+\log f))$時間で求められる． □

3 ソート点凸包問題を解く並列アルゴリズム

ここでは，平面上の点集合$S = \{s_1, s_2, \ldots, s_n\}$が（$x$座標に関して）ソートされた系列として与えられたときに，Sの凸包CH(S)を求める並列アルゴリズムを示す．Sの要素が全て一本の直線上にあるとき，CH(S)をSとする．他のとき，s_1,s_nそれぞれは，x座標が最小，最大であるので，$CH(S)$の頂点である．この2点により，$CH(S)$が2つの部分，$CH(S)$の上部境界UH(S)（時計順にs_1から，s_nまでの頂点系列）と$CH(S)$の下部境界$LH(S)$（時計順にs_nからs_1までの頂点系列）に分けられる．$LH(S)$は$UH(S)$と同様にして求めることができるので，以下では，$UH(S)$を求める方法についてのみ説明する．

3.1 アルゴリズムの概略

ここでは，アルゴリズムの概略を示す．このアルゴリズムをEREW,CRCWそれぞれのモデルで実現する方法については，3.3節で述べる．

{アルゴリズムSP}

[入力] 配列に格納された点集合S．ただし，各$i(1 \le i < n)$について，($S[i]$のx座標) $\le$ ($S[i+1]$のx座標)とする．

[出力]$CH(S)$の上部境界$UH(S)$．ただし，$UH(S)$を以下の双方向リストの形式で表す．$UH(S)$を記述するため，配列S，及び，配列I,L,Rを使う．$S[i]$が$UH(S)$の頂点のとき，$I[i] = 1$，他のとき，$I[i] = 0$である．$S[i]S[j]$が$UH(S)$の線分のとき（ただし，$i < j$とする），$L[j] = i$,$R[i] = j$である．

[方法]δまでの$n^{1/3}$分割統治法を用いてSの部分配列$S[first..last]$の上部境界$UH(S[first..last])$を求める方法を示す．ここでδはアルゴリズムのパラメタで，実行時には適当な値に設定される．この上部境界のデータ構造は次節で述べる．$UH(S)$は$first = 1, last = n$とすることによって求められる．ただし，$UH(S)$が次節で述べるデータ構造で求められるので，最後に，これを双方向リストに書き直す．

基底段階：

（ステップ1）$\hat{n} := first - last + 1$,$\hat{S}$を$S[first..last]$とする．$\hat{n} \le \delta$とき，3.3節で述べる方法で$UH(\hat{S})$を求める．

再帰段階：

（ステップ2）$n > \delta$とき，$\hat{S}$を$\lceil \hat{n}^{1/3} \rceil$個の同じサイズの部分配列$S_i$（$1 \le i \le \lceil \hat{n}^{1/3} \rceil$）に分ける．$UH(S_i)$を並列に再帰的に求める．

（ステップ3）$UH(S_i)$（$1 \le i \le \lceil \hat{n}^{1/3} \rceil$）を次のようにマージして$UH(\hat{S})$を求める．

(1) 各$UH(S_i)$に対して，$UH(S_j)$（$1 \le i < j \le \lceil \hat{n}^{1/3} \rceil$）との上側共通接線（簡単のため，以下では，単に共通接線と呼ぶ）T_{ij}を求める．

(2) 各k（$1 \le k \le \hat{n}^{1/3}$）に対して，共通接線の集合$\{T_{ik} | 1 \le i < k\}$における傾斜度（直線の傾斜度はその直線を時計まわりにy軸に重なるまで回転する角度と定義する）が最小の共通接線L_k（L_kの$UH(S_k)$における接点を$S[r1(k)]$，他方の接点を$S[f(k)]$とする）と共通接線の集合$\{T_{kj} | k < j \le \lceil \hat{n}^{1/3} \rceil\}$の中で傾斜度が最大の共通接線$R_k$（$R_k$の$UH(S_k)$における接点を$S[r2(k)]$，他方の接点を$S[g(k)]$とする）を求める．

(3) 各$UH(S_i)$に対して，$S[r1(i)]$が$S[r2(i)]$の左にある（つまり，$r1(i) < r2(i)$）とき，$S[r1(i)]$と$S[r2(i)]$の間の頂点が$UH(\hat{S})$の頂点である．また，そのとき，$UH(\hat{S})$において，$S[r1(i)]$の左の頂点は$S[f(i)]$，$S[r2(i)]$の右の頂点は$S[g(i)]$である． □

3.2 データ構造

アルゴリズムSPを再帰的に繰り返して実行するたびに，上部境界の共通接線を求める．この共通接線は，CREW-ACTAN, 或いは，EREW-ACTANを用いて求めることができるが，このためには，上部境界は配列に格納されている必要がある．ところが，上部境界を配列に格納するためには，大量のデータを移動させる（配列に格納し直す）必要があり，効率が悪い．そこで，次の性質を利用して，上部境界を表すために配列に類似したデータ構造を導入する．簡単のために，以下では，$n = \delta^{3^x}$とし，部分配列$S[(i-1)\delta+1..i\delta]$,$I[(i-1)\delta+1..i\delta]$（$1 \le i \le n/\delta$）それぞれを$S_{\delta(i)}$,$I_{\delta(i)}$と記述する．

性質 3 ソートされた平面点集合$S[1..n]$を$S_{\delta(1)}$,$S_{\delta(2)}$,…, $S_{\delta(n/\delta)}$に分け，$S_{\delta(i)}$の上部境界を$UH(S_{\delta(i)}) = S[i_1], S[i_2], \ldots, S[i_{k_i}]$（$1 \le i \le n/\delta$）とする．このとき，$S[u..v]$,($1 \le u < v \le n$, $v - u + 1 \ge \delta$)の上部境界$UH(S[u..v])$について，次の性質が成り立つ．$UH(S_{\delta(i)})$の頂点$S[i_f]$と$S[i_g]$が$UH(S[u..v])$の頂点なら，その間の$UH(S_{\delta(i)})$の頂点$S[i_x]$（$f \le x \le g$）も$UH(S[u..v])$の頂点である． □

Sの上部境界を求めるアルゴリズムSPの各再帰段階では，Sのある部分配列Sの上部境界を求める．性質3より，各$UH(S_{\delta(i)})$（$1 \le i \le n/\delta$）において，$UH(\hat{S})$に属する頂点はすべて連続しているので，最初の頂点$S[t(i)]$と最後の頂点$S[b(i)]$を表す2つのポインタ$t(i)$,$b(i)$を用いれば，$UH(S)$を$UH(S_{\delta(i)})$（$1 \le i \le n/\delta$）で表すことができる．よって，上部境界のデータ構造を次のようにする．

(1) 配列$I[1..n]$,$L[1..n]$,$R[1..n]$を用いて，基底段階で求める$UH(S_{\delta(i)})$（$1 \le i \le n/\delta$）を表す．つまり，$S_{\delta(i)}$の点$S[u]$が$UH(S_{\delta(i)})$の頂点なら，$I[u] = 1$，他のとき，$I[u] = 0$とする．また，$(S[u], S[v])$（$u < v$）が$UH(S_{\delta(i)})$の辺なら，$R[u] = v$,$L[v] = u$とする．

(2) ある再帰段階でアルゴリズムSPが処理している上部境界$UH(S)$を$UH(S_{\delta(i)})$（$1 \le i \le n/\delta$）と変数$t(i)$,$b(i)$で表

す．ここで，$S[t(i)],S[b(i)]$ は $UH(\hat{S})$ に現れる $UH(S_{\delta(i)})$ の最初と最後の頂点である．基底段階終了時に，$UH(S_{\delta(i)})=S[i_1],S[i_2],\ldots,S[i_{k_i}]$ なので，$t(i)=i_1$, $b(i)=i_{k_i}$であり，再帰段階を繰り返すにしたがって，$t(i)$ は増加し，$b(i)$ は減少する．そして，上部境界 $U(\hat{S})$ に属する $UH(S_{\delta(i)})$ の頂点がなくなったとき，そのことを示すため，特別の記号$\emptyset$を用いて，$t(i)=b(i)=\emptyset$ とする．

3.3 アルゴリズム SP の実現

次に，提案したデータ構造を用いてアルゴリズム SP の各モデルでの実現を考える．つまり，アルゴリズム SP のステップ 1，ステップ 3 の実現を考える．

まず，アルゴリズム SP のステップ 1 の実現を考える．

•ステップ 1 の実現

(1) 1 プロセッサ，$O(\delta)$ 時間での実現（EREW，CREW）

各 $UH(S_{\delta(i)})$ $(0\leq i\leq n/\delta)$ を逐次アルゴリズムで $O(\delta)$ 時間で求める．よって全ての $UH(S_{\delta(i)})$ $(0\leq i\leq n/\delta)$ は n/δ プロセッサ，$O(\delta)$ 時間で求められる．

(2) δ^3プロセッサ，定数時間での実現（CRCW）

各 $UH(S_{\delta(i)})$ $(0\leq i\leq n/\delta)$ を，CRCW PRAM の上で，次のようにδ^3プロセッサ，$O(1)$ 時間で求める．

(a) 配列 Iの各要素の値を 0 にする．

(b) $S_{\delta(i)}$の各点 $S[(i-1)\delta+j]$ $(1\leq j\leq\delta)$ について，線分 $(S[(i-1)\delta+x],S[(i-1)\delta+j])$ $(1\leq x<j)$ の中で傾斜度が最小の線分 L_j，線分 $(S[(i-1)\delta+j],S[(i-1)\delta+y])$ $(j<y\leq\delta)$ の中で傾斜度が最大の線分 R_j を求める．時計回りの順で線分 L_j から線分 R_j までの角度が 180°以上ならば，L_j，R_j がともに $UH(S_{\delta(i)})$ の辺である．このとき，$L_j=(S[(i-1)\delta+x^*],S[(i-1)\delta+j])$, $R_j=(S[(i-1)\delta+j],S[(i-1)\delta+y^*])$ とすると，$I[j]=1,L[j]=x^*,R[j]=y^*$ にする．

δ個の要素の集合の最小値と最大値が CRCW 上でδ^2プロセッサ，$O(1)$ 時間で求められる[8]ので，$UH(S_{\delta(i)})$ は，δ^3プロセッサ，$O(1)$ 時間で求められる．

• ステップ 3 (1) の実現

アルゴリズム SP のステップ 3 (1)，つまり，各 i,j $(1\leq i<j\leq n^{1/3})$ について上部境界 $UH(S_i)$ と $UH(S_j)$ の共通接線を求める部分の実現を考える．アルゴリズム EREW-ACTAN(f) とアルゴリズム CREW-ACTAN(f) を少々修正すれば本データ構造の上でも動くので[14]，凸多角形対の数 x を $n^{1/3}$にすると，次の補題が成り立つ．

補題 1 全ての i,j $(1\leq i<j\leq n^{1/3})$ について上部境界 $UH(S_i)$ と $UH(S_j)$ の共通接線 T_{ij}は

(1) EREW で$\hat{n}^{2/3}f$プロセッサ，$O((\log^2\hat{n}/\log^2 f+1)(\log n+\log f))$ 時間で求められる．

(2) CREW で$\hat{n}^{2/3}f$プロセッサ，$O((\log\hat{n}/\log f)^2+1)$ 時間，

□

• ステップ 3 (2) の実現

アルゴリズム SP のステップ 3 の (2) の実現を考える．各 L_i (${\hat{n}}^{1/3}$個の接線の中に傾斜度が最小の接線) $(1\leq i\leq\hat{n}^{1/3})$ は，EREW の上で$\hat{n}^{1/3}/\log n$プロセッサ，$O(\log n)$ 時間[8]，CRCW の上で$\hat{n}^{2/3}$プロセッサ，$O(1)$ 時間[8]で求められる．従って，アルゴリズム SP のステップ 3 の (2) は，EREW の上で，$\hat{n}^{2/3}/\log\hat{n}$プロセッサ，$O(\log\hat{n})$ 時間，CRCW の上で，$\hat{n}$ プロセッサ，$O(1)$ 時間で実現できる．

•ステップ 3 (3) の実現

アルゴリズム SP のステップ 3 の (3) の実現を考える．各 S_iに対して，kプロセッサ，$O(1)$ 時間で以下のことをする．

(1) $UH(S)$ に $UH(S_i)$ の頂点がある場合．

線分 $L_i=(S[f(i)],S[r1(i)]),R_i=(S[r2(i)],S[g(i)])$ が $UH(S)$ の辺である．$S[f(i)],S[r1(i)],S[r2(i)],S[g(i)]$ それぞれを $UH(S_{\delta(u)}),UH(S_{\delta(v)}),UH(S_{\delta(w)}),UH(S_{\delta(x)})$ の頂点とする（$u<v\leq w<x$.u,v,w,x は共通接線を求める時点で得られる）．このとき，次のように実行する．

(a) $L[r1(i)]:=f(i),R[f(i)]:=r1(i),L[g(i)]:=r2(i)$, $R[r2(i)]:=g(i),t(v):=r1(i),b(w):=r2(i)$ とする．

(b) 各 y $(v<y<w)$ について，その時点で処理している上部境界に$UH(S_{\delta(y)})$の頂点がないので，$t(y):=\emptyset,b(y):=\emptyset$ とする．

(2) $UH(S)$ に $UH(S_i)$ の頂点がない場合．

$UH(S_i)=UH(S[(i-1)\hat{n}^{2/3}+1..i\hat{n}^{2/3}])=UH(S_{\delta(a)}\cup S_{\delta(a+1)}\cup\cdots\cup S_{\delta(b)})$ $(a=(i-1)\hat{n}^{2/3}/\delta,\ b=i\hat{n}^{2/3}/\delta-1)$ より，各 y $(a\leq y\leq b)$ に対して，$t(y):=\emptyset,b(y):=\emptyset$ とする．

以上の (1),(2) は，同時書き込み，同時読み出しがないので，EREW PRAM，CRCW PRAM のどちらの上でも$\hat{n}/\delta$プロセッサ，$O(1)$ 時間で実現できる．

•アルゴリズム全体の計算量

アルゴリズム EREW-SP,CRCW-SP,CRCW-SPCT の各ステップの計算量を表 2 にまとめておく．ここでは，EREW モデルに対して得られた計算量は，そのまま CREW モデルの上，CREW モデルに対して得られた計算量は，そのまま CRCW モデルの上でも成り立つことを利用している．

表 2 アルゴリズムAの解析

ステップ(一回の実行) ＼ アルゴリズム／モデル		EREW-SP EREW	CRCW-SP CREW	CRCW-SPCT CRCW
1	プロセッサ数	1	1	δ^3
	時間計算量	$O(\delta)$	$O(\delta)$	$O(1)$
3 (1)	プロセッサ数	$\hat{n}^{2/3}$	$\hat{n}^{2/3}$	$\hat{n}^{2/3}$
	時間計算量	$O((\log\hat{n}/\log f)+1)^2$ *$(\log\hat{n}+\log f)$	$O((\log\hat{n}/\log f)+1)^2$	$O((\log\hat{n}/\log f)+1)^2$
3 (2)	プロセッサ数	$\hat{n}^{2/3}/\log\hat{n}$	$\hat{n}/\delta$	$\hat{n}$
	時間計算量	$O(\log\hat{n})$	$O(\delta)$	$O(1)$
3 (3)	プロセッサ数	$\hat{n}/\delta$	$\hat{n}/\delta$	$\hat{n}/\delta$
	時間計算量	$O(1)$	$O(1)$	$O(1)$

(1) アルゴリズム EREW-SP の計算量

$\delta=\log n,f=n^{1/3}/\log n$ とするとき，プロセッサ数は $n/\log n$ で，計算時間 $T(n)$ は次の式で表される．

$$T(\hat{n})=\begin{cases}T(\hat{n}^{2/3})+O((\log^2\hat{n}/\log^2(\hat{n}^{1/3}/\log n)+1)\\ \quad(\log\hat{n}+\log(\hat{n}^{1/3}/\log n)), & \hat{n}>\log n\\ O(\log n), & n\leq\log n\end{cases}$$

よって $T(n)=O(\log n)$ である．

(2) アルゴリズム CRCW-SP の計算量

$\delta=\log\log n,f=\hat{n}^{1/3}/\log\log n$ とするとき，プロセッサ数は $n/\log\log n$ で，計算時間 $T(n)$ は次の式で表される．

$$T(n)=\begin{cases}T(\hat{n}^{2/3})+O(\log^2\hat{n}/\log^2(\hat{n}^{1/3}/\log\log n)+1),\\ \qquad n>\log\ logn\\ O(\log\log n), & n\leq\log\log n,\end{cases}$$

よって，$T(n) = O(\log\log n)$ である．

(3) アルゴリズム CRCW-の計算量

CRCW で $\delta = n^{\epsilon/2}, f = \hat{n}^{1/3}$ とするとき，プロセッサ数は $n^{1+\epsilon}$ で，計算時間 $T(n)$ は次の式で表される．

$$T(\hat{n}) = \begin{cases} T(\hat{n}^{2/3}) + O(1), & \hat{n} > n^{\epsilon/2} \\ O(1), & \hat{n} \leq n^{\epsilon/2} \end{cases}$$

よって，$T(n) = O(1)$ である．

最後に，求められた上部境界 $UH(S)$ を双方向リストに書き直す．任意の i $(1 \leq i \leq n)$ に対して，$S[i]$ は $S[\delta(i')]$ $(i' = \lceil (i/\delta) \rceil - 1)$ の要素である．$t(i') \neq \emptyset$ とき，$t(i') \leq i \leq b(i')$ かつ $I(i) = 1$ であれば，$I(i) = 1$ とし，その以外のとき，$I(i) = 0$ とする．各 $S_{\delta(i)}$ に 1 つのプロセッサを割り当てれば，$S_{\delta(i)}$ 内の処理が $O(\delta)$ 時間でできる．

定理 4 ソート点凸包問題（リスト構造形式）は

(1) EREW で，$n/\log n$ プロセッサ，$O(\log n)$ 時間で解ける．

(2) CRCW で，$n/\log\log n$ プロセッサ，$O(\log\log n)$ 時間で解ける．

(3) CRCW で，$n^{1+\epsilon}$ プロセッサ，$O(1)$ 時間で解ける． □

4 多角形凸包問題を解く最適並列アルゴリズム

単純 n 角形 P が時計順の頂点系列 $p_1, p_2, \ldots, p_n$ で与えられているとき，P の凸包を求める問題を考える．P の頂点 p_i $(1 \leq i \leq n)$ の x 座標を $x(p_i)$，y 座標を $y(p_i)$ と表す．一般性を失うことはなく，以下では，p_1 を P における x 座標の最小の頂点と仮定する．

単純多角形 P の上部境界については，次の性質が成り立つ．

性質 4 単純 n 角形 P $(= p_1, p_2, \ldots, p_n)$ の上部境界 $UH(P)$ を時計順の頂点系列 $p_{h_1}, p_{h_2}, \ldots, p_{h_k}$ とするとき，$h_i < h_{i+1}$ $(1 \leq i \leq k-1)$ である． □

単純多角形 P の二つの部分頂点系列 $P1$ と $P2$ それぞれを $P1 = p_{i_1}, p_{i_1+1}, \ldots, p_{j_1}$，$P2 = p_{i_2}, p_{i_2+1}, \ldots, p_{j_2}$ $(i_1 < j_1 < i_2 < j_2)$ とする．$UH(P1)$ と $UH(P2)$ が一本の垂直線で左と右に分けられるとき，UH(P1) と UH(P2) の上側共通接線は唯一である．$UH(P1)$ と $UH(P2)$ が一本の垂直線で左と右に分けらないとき，次の性質より，高々2 本の上側共通接線があり，そのうちの高々1 本が $UH(P)$ の辺になる．

性質 5 *(1)* $UH(P1)$ と $UH(P2)$ の上側共通接線は高々2 本である．

(2) $UH(P1)$ と $UH(P2)$ の上側共通接線が 2 本のとき，その 2 本の上側共通接線それぞれを $T_1 = (p_a, p_b)$，$T_2 = (p_c, p_d)$ $(p_a, p_c$ が $P1$ の接点，p_b, p_d が $P2$ の接点$)$ とするとき，少なくとも次のいずれかが成り立つ．

(a) $x(p_a) > x(p_b)$

(b) $x(p_c) > x(p_d)$

（証明）$UH(P1)$ と $UH(P2)$ が一本の垂直線で左と右に分けられるとき，上側共通接線が 1 本しかないのは明らかである．以下では，$UH(P1)$ と $UH(P2)$ が一本の垂直線で左と右に分けられないときについて考える．$UH(P1)$ と $UH(P2)$ が交差しない場合，上側共通接線が高々2 本がある．そのとき，性質 *5(2)* が明らかに成り立つ（図 *2(a)*）．従って，以下では，$UH(P1)$ と $UH(P2)$ が交差する場合についてのみ考える．

まず，$UH(P1)$ と $UH(P2)$ の交点数が 2 以下であることを示す．$UH(P1)$ と $UH(P2)$ の交点数が 3 以上とし，任意の 3 つの交点を q_1, q_2, q_3 とする．ただし，$x(q_1) < x(q_2) < x(q_3)$ とする．そのとき，$UH(P1)$ の辺 $l^{P1}(i) = (p_{u(i)}, p_{v(i)})$ と $UH(P2)$ の辺 $l^{P2}(i) = (p_{s(i)}, p_{t(i)})$ $(1 \leq i \leq 3)$ が q_i で交差しているとし，$UH(P1)$ において，6 つの頂点が時計順に $p_{u(1)}, p_{v(1)}, p_{u(2)}, p_{v(2)}, p_{u(3)}, p_{v(3)}$ の順で現れる（6 つの頂点が異なると仮定する．場合によって，$P_{v(1)}$ と $P_{u(2)}$，あるいは，$P_{v(2)}$ と $P_{u(3)}$ が同じ頂点である．このときも同様である．*UH(P2)* についても同様）とし，$UH(P2)$ において，6 つの頂点が時計順に $p_{s_1}, p_{t_1}, p_{s_2}, p_{t_2}, p_{s_3}, p_{t_3}$ の順で現れるとする（図 2(b)）．以上の 6 つの辺それぞれを無限遠へ延ばして得られる直線それぞれを $l_\infty^{P1}(i)$ と $l_\infty^{P2}(i)$ $(1 \leq i \leq 3)$ とする．直線 $l_\infty^{P1}(1)$ と直線 $l_\infty^{P1}(2)$ との交点を p，直線 $l_\infty^{P1}(2)$ と直線 $l_\infty^{P1}(3)$ との交点を p'，直線 $l_\infty^{P2}(1)$ と直線 $l_\infty^{P2}(2)$ との交点を q，直線 $l_\infty^{P2}(2)$ と直線 $l_\infty^{P2}(3)$ との交点を q' とするとき，凸チェーン $P1' = p_{u_1}, p_{v_1}, p, p_{u_2}, p_{v_2}, p', p_{u_3}, p_{v_3}$ が $UH(P1)$ の上にあり，凸チェーン $P2' = p_{s_1}, p_{t_1}, q, p_{s_2}, p_{t_2}, q', p_{s_3}, p_{t_3}$ が $UH(P2)$ の上にある．凸チェーン $P1'$ と凸チェーン $P2'$ について考える．以下では，一般性を失うことなく，p の y 座標 $> q$ の y 座標と仮定する．このとき，p' の y 座標 $< q'$ の y 座標である．P において，頂点 p_{u_2} から頂点 p_{v_2} までのパスを $path_1$，頂点 p_{s_2} から頂点 p_{t_2} までのパスを $path_2$ とする．このとき，パス $path_1$ は辺 $l^{P1}(2)$ の下にあり，パス $path_2$ は辺 $l^{P2}(2)$ の下にある．また，単純多角形 P の辺と辺は交差しないので，この 2 つのパスは必ず一方が他方の上にある．まず，パス $path_1$ がパス $path_2$ の上にあるとする（図 2(c)）．そのとき，頂点 p_{u_2} と頂点 p_{u_3} の間のパスは，パス $path_2$ と交差し，P の辺と辺が交差することになり，矛盾を生じる．パス $path_2$ がパス $path_1$ の上にある場合も同様に矛盾を生じる．従って，$UH(P1)$ と $UH(P2)$ の交点数は 2 以下である．

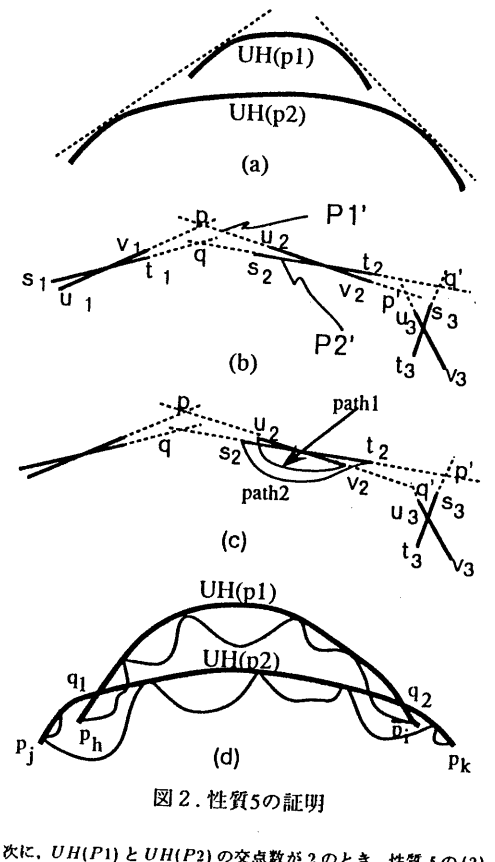

図 2．性質5の証明

次に，$UH(P1)$ と $UH(P2)$ の交点数が 2 のとき，性質 5 の (2) が成り立つことを示す．交点数が 1 のときも同様に証明できる．$UH(P1)$ と $UH(P2)$ の 2 つの交点を q_1, q_2 とし，$x(q_1) < x(q_2)$

とする．また，一般性を失うことなく，交点 q_1 と交点 q_2 の間の $UH(P1)$ の部分は $UH(P2)$ の上にあると仮定する．$UH(P1)$ の最初と最後の頂点それぞれを p_h と p_i，$UH(P2)$ の最初と最後の頂点それぞれを p_j と p_k とするとき，Pにおいて，p_h と p_i の間のパスは，必ず，p_j と p_k の間のパスの上にある（図 2(d)）．そのとき，$UH(P1)$ と $UH(P2)$ の上側共通接線は 2 つある．左の一つを (p_a, p_b) $(p_a \in P1, p_b \in P2)$ とすると，$x(p_a) > x(p_b)$ が成り立つ．
□

性質 5 の (2)(a) を満たす共通接線 T_1 の 2 つの端点 p_a, p_b $(p_a \in P1, p_b \in P2, x(p_a) > x(p_b))$ が $UH(P)$ の頂点であると仮定する．このとき，$UH(P)$ の上で時計順に頂点 p_a が頂点 p_b の後にある．これは性質 4 と矛盾する．従って，T_1 は $UH(P)$ の辺にならない．性質 5 の (2)(b) を満たす共通接線 T_2 についても同様である．

以上の性質より，多角形凸包問題を解く並列アルゴリズム POL には，ソート点凸包問題を求めるアルゴリズム SP を次のように変更すればよい．

(1) まず，アルゴリズム SP の基底段階で用いるソート点凸包問題を解く逐次アルゴリズムを多角形凸包問題を解く逐次アルゴリズムに置き換える（定数時間で多角形凸包問題を解く場合には，基底段階を定数時間で解くアルゴリズムをそのまま用いる）．

(2)2 章のアルゴリズムで上部境界 の上側共通接線を求めるとき，求められた共通接線が性質 5 の (a) 或いは (b) を満たせば，その共通接線を捨てる．残った共通接線は高々1 本である．従って，アルゴリズム SP の再帰段階の他の部分はそのまま用いられる．

アルゴリズム POL を，アルゴリズム SP と同様に，EREW,CRCW それぞれのモデルで実現することによって，アルゴリズム EREW-POL,CRCW-POL,CRCW-POLCT が得られる．

定理 5 多角形凸包問題（リスト構造形式）は

(1) EREW 上で，$n/\log n$ プロセッサ，$O(\log n)$ 時間で解ける．

(2) CRCW の上で，$n/\log \log n$ プロセッサ，$O(\log \log n)$ 時間で解ける．

(3) CRCW の上で，$n^{1+\epsilon}$ プロセッサ，$O(1)$ 時間で解ける．　□

5 アルゴリズムの最適性

●ソート点凸包問題

n 個の整数の集合 $I = i_1, i_2, ..., i_n$ の最大値，最小値を求める問題を，ソート点凸包問題を利用して，次のように解ける．x 座標によってソートされた平面点集合 $L = \{(-1,0), (0,i_1), (0,i_2), \ldots, (0,i_n), (1,0)\}$ に対して，凸包 $CH(L)$ の識別形式を求める．$CH(L)$ が 4 つの点 $(-1,0), (0,i_{k_1}), (0,i_{k_2}), (1,0)$ からなるとき，i_{k_1} と i_{k_2} は I の最大値と I の最小値である．$CH(L)$ が 3 つの点 $(-1,0), (0,i_k), (1,0)$ からなるとき，$(0,i_k)$ は I の最大値（あるいは，最小値）である．そのとき，$L' = \{(-1,i_k), (0,i_1), (0,i_2), \ldots, (0,i_n), (1,i_k)\}$ に対して，凸包 $CH(L')$ の識別形式を求める．凸包 $CH(L')$ が 3 つの点 $(-1,i_k), (1,i_k), (0,i_{k'})$ からなり，このとき，$(0,i_{k'})$ は I の最小値（あるいは，最大値）である．つまり，凸包 $CH(L)$ と凸包 $CH(L')$ の識別形式を求めれば，I の最大と最小値は L の各要素に 1 個のプロセッサを割り当て，$O(1)$ 時間で判定できる．従って，I の最小値を求める最適並列アルゴリズム（コストは $O(n)$）の時間計算量の下界は，ソート点凸包問題の識別形式を求める最適アルゴリズムの時間計算量の下界である．I の最小値を求める最適並列アルゴリズムの時間計算量は，CREW では，$\Omega(\log n)^{(2)}$ で，CRCW では，$\Omega(\log \log n)^{(10)}$ であることから，本稿の点凸包問題のリスト構造形式を求めるアルゴリズム EREW-SP, CRCW-SP は最適並列アルゴリズムである．

●多角形凸包問題

ソート点凸包問題は次のように多角形凸包問題に帰着できる．

x 座標でソートされた n 個の点の点集合 $S = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n)\}$ に対して，$((x_i,y_i),(x_{i+1},y_{i+1}))$ $(1 \le i \le n-1)$，$((x_n,y_n),(x_1,-\infty)), ((x_1,-\infty),(x_1,y_1))$ それぞれを辺と見なせれば，$S' = \{(x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n), (x_1,-\infty)\}$ が単純多角形になる．よって，本稿の多角形凸包問題のリスト構造形式を求めるアルゴリズム EREW-POL,CRCW-POL は最適並列アルゴリズムである．

6 むすび

ソートされた平面点集合の凸包と単純多角形の凸包を求める，以下の並列アルゴリズムを提案した．(1) EREW 上の $n/\log n$ プロセッサ，$O(\log n)$ 時間のアルゴリズム．(2) CRCW 上の $n/\log\log n$ プロセッサ，$O(\log \log n)$ 時間のアルゴリズム．(3) CRCW 上の $O(n^{1+\epsilon})$ プロセッサ，$O(1)$ 時間のアルゴリズム．アルゴリズム (1) と (2) が最適であり，アルゴリズム (3) は，プロセッサ数が多く使えるような状況では，アルゴリズム (1), (2) より高速である．

参考文献

(1) A.Aggarwal,B.Chazelle,L.Guibas,C.O'Dunlaing and C.Yap: "Parallel computational geometry",Algorithmica,3,pp.293-327(1988).

(2) S.G.Akl:"A Constant-Time Parallel Algorithm for Computing Convexs Hulls",Bit,Vol,22,pp.130-134(1982).

(3) M.J.Atallah and D.C.Chen:"An optimal parallel algorithm for the visibility of a simple polygon from a point",Proc. of the Fifth Annual ACM Symposium on Computational Geometry, pp. 114-123(1989).

(4) M.J.Atallah and M.T.Goodrich:"Efficient parallel solutions to some geometric problems",Journal of Parallel and Distributed Computing,3,pp.492-507(1986).

(5) M.J.Atallah and M.T.Goodrich:"Parallel algorithms for some functions of two convex polygons",Algorithmica,3,pp.535-548(1988).

(6) R.Cole:"Faster optimal paralell prifix sums and list ranking", Information and Control,81,pp.334-352(1989).

(7) P-O.Fjällström,J.Katajainen,C.Levcopoulos and O.Petersson:"A sublogarithmic convex hull algorithm", Bit,30,pp.378-384(1990).

(8) A.Gibbons and W.Rytter:"Efficient parallel algorithms", Camcridge University Press(1988).

(9) M.T.Goodrich:"Finding the convex hull of a sorted point set in parallel",Information and Computation,81,pp.334-352(1989).

(10) F.P.Preparata and M.L.Shamos:"Computational Geometry:An Introduction ",Springer-Verlag(1985).

(11) L.G.Valiant:"Parallelism in comparison problem",SIAM J. Comput.,4,3,pp.348-355(1975).

(12) R.Miller and Q.F.Stout:"Efficient parallel convex hull algorithms",IEEE Trans. Comput.,Vol.37,No.12,pp.1605-1668(1988).

(13) 榎原，中野，中西:"凸包問題に対する並列アルゴリズム", 信学会情報・システム部門全国大会",59(1989)

(14) 陳慰，中野浩嗣，増澤利光，辻野嘉宏，都倉信樹，単純多角形における 2 点間の最短経路を求める最適並列アルゴリズム，信学技報,COMP90-88(1991-1).