

Voronoi diagram 構成のための並列アルゴリズム

阿部 達朗*・井上 克司**・高浪 五男**・谷口 弘**

A Parallel Algorithm for the Construction of Voronoi Diagrams

Tatsuro ABE, Katsushi INOUE, Itsuo TAKANAMI and Hiroshi TANIGUCHI

Abstract

This paper presents a new algorithm for the construction of Voronoi diagrams on a shared memory parallel computer, where both concurrent reads and concurrent writes are allowed, but all the processors that simultaneously try to write in the same memory cell must write the same value.

The algorithm is a parallel version of the sequential algorithm (for the construction of Voronoi diagrams) of Shamos and Hoey. The algorithm, for n input points, runs in $O(\log^3 n)$ time with only n processors.

1. まえがき

近年、幾何学上の種々の問題を計算機モデルを用いて解法する計算幾何学 (Computational Geometry) に注目が集まっている。計算幾何学は、パターン認識、VLSI における回路設計、コンピュータ・グラフィクスなどの様々な分野に应用されており、今後、益々その重要性が高まってくるものと思われる¹⁾。

幾何学上の種々の問題を解決するために提案されているアルゴリズムの多くは、逐次型計算機モデルを対象に設計されており、一般に大きな計算時間を必要とする。

一方、最近、計算の高速化を目指して様々な並列計算機モデルが提案され、種々の組合せ問題 (例えば、グラフの連結性を判定する問題、ソーティング、マーキングなど) をこれらの並列計算機モデル上で解くためのアルゴリズムが開発されている^{2)~5)}。幾何学上の問題を並列計算機モデル上で解決するためのアルゴリズムの研究も行われ始めているが、まだ十分な成果は得

られていない^{6)~8)}。

本論文では、計算幾何学における最も重要な問題の1つである Voronoi diagram を取り上げ、任意に与えられた平面上の N 個の点集合の Voronoi diagram を N 個のプロセッサを用いて構成するための高速な ($O(\log^3 N)$) 並列アルゴリズムを与える。文献[6]においても $O(N)$ 個のプロセッサを用いて平面上の N 個の点集合の voronoi diagram を構成するための $O(\log^3 N)$ の並列アルゴリズムが与えられているが、本論文で述べる並列アルゴリズムは、文献[1]で与えられている Voronoi diagram 構成のための逐次型アルゴリズムを並列化したものである。

本論文で用いる並列計算機モデルは、複数のプロセッサが同一のメモリセルに同時にアクセスして読み書き可能な共有メモリを有する PRAM (Parallel Random Access Machine) である。但し、複数のプロセッサが同一のメモリセルに同時に書き込むときには、全く同一の内容を書き込む動作のみが許される²⁾。

2. Voronoi diagram 構成のための並列アルゴリズム

ここではまず、平面上の N 個の点の集合 $S = \{P_1, P_2, \dots, P_N\}$ が与えられたときの、その voronoi diagram

*東芝㈱

**電子工学科

(これを $VOR(S)$ と記す) を N 個のプロセッサの PRAM で構成する並列アルゴリズムの概要を与える。図 1 に 16 個の点からなる Voronoi diagram を示す。Voronoi diagram $VOR(S)$ は、平面の N 個の同値類 (S の 1 つの点が一つの同値類に対応する) への分割である。点 $P_i \in S$ に対する同値類を $V(P_i)$ と記し、これを点 P_i の Voronoi polygon と呼ぶ。更に、Voronoi diagram の各頂点は Voronoi vertex と呼ばれ、各線分は Voronoi edge と呼ばれる。Voronoi diagram の全ての Voronoi vertex は、diagram の 3 つの Voronoi edge の共通接点であり、 S の 3 つの点によって定義される円の中心である。Vertex v に対して上記の円を $C(v)$ と示す。明らかに、 S の Voronoi diagram の全ての vertex v に対して、円 $C(v)$ の内部に S の他の点は含まれない。Voronoi diagram の定義の詳細は文献 [1] を参照して頂きたい。

本論文で与える $VOR(S)$ 構成のための並列アルゴリズムは、Shamos と Hoey によって提案された $O(N \log N)$ の逐次型アルゴリズムを並列化したものである。この逐次型アルゴリズムでは、垂直線 L によって S を等しい大きさの 2 つの部分集合 S_1, S_2 に分割して、それぞれの Voronoi diagram $VOR(S_1)$ と $VOR(S_2)$ を構成し、これらを併合する分割統治法 (divide and conquer) を用いている。併合にかかる時間は線形時間である。ここで重要なのは、併合の方法であり、彼らは幾何学的作図を用いて、併合を線形時間で行っている。即ち、 S の分割 $\{S_1, S_2\}$ を与えたとき、 $VOR(S)$ の Voronoi polygon $V(P_i)$ と $V(P_k)$ ($P_i \in S_1, P_k \in S_2$) によって共有される Voronoi edge の集合 $\sigma(S_1, S_2)$ を作成し、これを併合に有効に利用している。 σ は Y 軸に関して single monotone chain である。 σ の定義の詳細は文献 [1] を参照して頂きたい。Fig. 2 に、Fig. 1 の点集合に対する $VOR(S_1)$ と $VOR(S_2)$ と $\sigma(S_1, S_2)$ を示す。

考案した並列アルゴリズムは次の 4 ステップに分かれる。

- (1) **ステップ 1** : S を X 座標に従ってソート (sort) する。

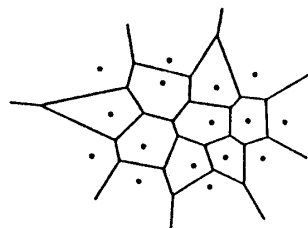


Fig. 1 Voronoi Diagram.

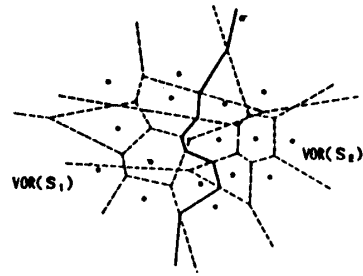


Fig. 2 $VOR(S_1), VOR(S_2)$ and chain σ .

- (2) **ステップ 2** : S を垂直線 $L (X = a)$ によって、等しい大きさの 2 つの部分集合 S_1, S_2 に分割する。
- (3) **ステップ 3** : それぞれ $VOR(S_1), VOR(S_2)$ を作る。
- (4) **ステップ 4** : $VOR(S)$ を得るために $VOR(S_1)$ と $VOR(S_2)$ を併合する。

N 個のプロセッサを用いて、 N 個の点集合 S の Voronoi diagram $VOR(S)$ を構成する本並列アルゴリズムの実行時間を $T(N)$ とする。ステップ 1 は文献 [5] のアルゴリズムを用いれば、 N 個のプロセッサを用いて $O(\log N)$ で実行可能であり、またステップ 2 は $O(1)$ で実行可能である。ステップ 3 の実行時間は、明らかに $T(N/2)$ である。次章以後で、ステップ 4 の実行時間は $O(\log^2 N)$ であることが示される。したがって、ある正の定数 $c > 0$ に対し、

$$T(N) = T(N/2) + c \log^2 N$$

を得る。これから、 $T(N) = O(\log^3 N)$ であることが示され、次の定理を得る。

【定理 1】 平面上の N 個の点の集合 S の Voronoi diagram $VOR(S)$ は、 N 個のプロセッサを用いた PRAM 上で $O(\log^3 N)$ の時間で構成され得る。

以後、ステップ 4 に焦点を合わせる。

3. ステップ 4 の詳細

本章では、上で述べたステップ 4 が $O(\log^2 N)$ で実行可能であることを示す。

【仮定 1】 集合 S の任意の 2 点は、 Y 軸に平行な同一直線上に存在しない。

存在してもしなくてもアルゴリズムの本質には何ら変わりはない。

【仮定 2】 集合 S の任意の 4 点は、同一円周上に存在しない。

もし、この仮定が真でないならば、voronoi diagram の定義に反し、Voronoi diagram は構成されない。

3.1 ステップ4の概要

ステップ4の入力は $VOR(S_1), VOR(S_2)$ である。但し、 $VOR(S_1)[VOR(S_2)]$ は、 $S_1[S_2]$ の各点の Voronoi polygon の集合として与えられる。各 Voronoi polygon は、AVL木⁹⁾の形で共有メモリに格納されている。AVL木はその Polygon の hull 上の各 vertex を時計回りに中間順 (inorder) に記憶している。特に、open Voronoi polygon の場合、無限の長さを持つ Voronoi edge の端点に相当する vertex を格納している AVL木中の頂点には、その Voronoi edge の方向情報を記憶させる。AVL木の各頂点には次の様な情報も記憶させる。

- (1) その頂点に対応する Voronoi vertex v_i の座標と v_i の次と前の (時計回りに見て) Voronoi vertex を記憶している (AVL木の) それぞれの頂点へのポインター。
- (2) 円 $C(v_i)$ 上の3つの (S_1 または S_2 中の) 点へのポインター。
- (3) その頂点の左部分木と右部分木に含まれる頂点の数。

ステップ4の出力は $VOR(S)$ で、 $VOR(S)$ は S 中の各点の Voronoi polygon の集合で与えられ、各 Voronoi polygon は、 $VOR(S_1), VOR(S_2)$ の場合と同様、AVL木で表現される。

ステップ4では、 $VOR(S_1)$ あるいは $VOR(S_2)$ における Voronoi vertex が $VOR(S)$ における Voronoi vertex として残されるか否かを調べ、残されない vertex を用いて $\sigma(S_1, S_2)$ を構成し、 $\sigma(S_1, S_2)$ から $VOR(S)$ における新たな Voronoi vertex を見つけ出す。 $VOR(S_1)[VOR(S_2)]$ における Voronoi vertex v が、 $VOR(S)$ の Voronoi vertex として残されるか否かは、円 $C(v)$ 中に $S_2[S_1]$ の点が含まれるか否かで決められ、もし、含まれなければ、残され、さもなければ、残されないことになる。

3.2 準備

まず、ステップ4のアルゴリズムの詳細を述べる前に2, 3の用意をしておく。

S_1, S_2 をステップ2で得られた S の分割とする ($S = S_1 \cup S_2$)。以後、混乱のない限り、 $VOR(S_1)$ あるいは $VOR(S_2)$ における点 $P_r (P_r \in S)$ の Voronoi polygon を $V(P_r)$ と記すことにする。

$CH(S)$ (S の Convex hull を $CH(S)$ と記す) を得るために $CH(S_1)$ と $CH(S_2)$ を併合する^{7,8)} ときの upper tangent point を P_{u1}, P_{u2} , lower tangent point を $P_{l1}, P_{l2} (P_{u1}, P_{l1} \in S_1, P_{u2}, P_{l2} \in S_2)$ としたとき、

$P_{u1}[P_{l2}]$ から時計回りに $P_{l1}[P_{u2}]$ までの $CH(S_1)[CH(S_2)]$ の hull point の集合を $A_1[A_2]$ とする。 $P_i \in A_1$ としよう。 $P_i \in A_2$ の場合についても同様である。 $V(P_i)$ は open boundary polygon で、半直線である2つの edge e_{ih}, e_{ih}' が存在する。 e_{ih}, e_{ih}' は、それぞれ A_1 に属する2点によって構成される Voronoi edge である。点 P_i を端点とし、 e_{ih}, e_{ih}' にそれぞれ平行な $V(P_i)$ の周と交わらない半直線を H_i, H_i' とする。議論の簡単のため、 $P_i \neq P_{u1}, P_{l1}$ とする。また、 e_{ih}, e_{ih}' とそれぞれ隣接している vertex を v_{ih}, v_{ih}' とする。

ステップ4によって作られる $VOR(S)$ における $V(P_i)$ は必ず close boundary polygon となる。つまり、 σ は必ず H_i, H_i' と交わる。今、 H_i について考える。 H_i' についても同様である。 H_i と交わる σ のある edge の延長線が線分 $\overline{P_i v_{ih}}$ と交わらないときがあり (このときは円 $C(v_{ih})$ の内部に S_2 の点を含まず、 v_{ih} は $VOR(S)$ の Voronoi vertex として残り、 σ の作成には利用されない)、この場合は σ の作成のために、 H_i を用いる。

$S_1[S_2]$ の点で、 $VOR(S_1)[VOR(S_2)]$ におけるその Voronoi polygon の一部が中心線 L より右側[左側]にも存在し、しかもその Voronoi polygon が $A_1[A_2]$ による hull と交わるような点の集合を $T_1[T_2]$ とする。

また、 $P_i \in T_1 [P_k \in T_2]$ に対して、中心線 L より左側の $V(P_i)[V(P_k)]$ の部分を $'V(P_i)['V(P_k)]$ 、右側の $V(P_i)[V(P_k)]$ の部分を $V'(P_i)[V'(P_k)]$ とする。即ち、 $V(P_i) = 'V(P_i) \cup V'(P_i) [V(P_k) = 'V(P_k) \cup V'(P_k)]$ となる。

更に、 $T_1[T_2]$ の点で、 $V'[V]$ が close boundary polygon である点の集合を $C_1[C_2]$ 、open boundary polygon である点の集合を $O_1[O_2]$ とする。

$O_1 \subset A_1 [O_2 \subset A_2]$ でもある。

$P_i \in C_1 [P_k \in C_2]$ で、 $V(P_i)[V(P_k)]$ の最も右側[左側]の vertex から、更に右側[左側]へ続いている X 軸に対して monotone な $VOR(S_1)[VOR(S_2)]$ における Voronoi edge の chain を $\delta(i) [\delta(k)]$ と記す。Fig. 3 に $\delta(K)$ の例を示す。

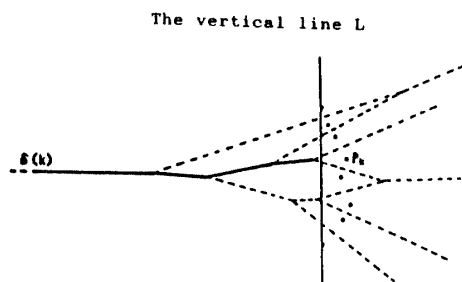


Fig. 3 chain $\delta(k)$.

3.3 ステップ4のアルゴリズム

$S = \{P_1, P_2, \dots, P_N\}$ の点をステップ1でX座標についてソートした結果,

$$X(P_1) < X(P_2) < \dots < X(P_N)$$

となっているものとする。但し、 $X(P_i)$ は、点 P_i のX座標を表す。認識番号 i のプロセッサは点 P_i をも担当する。各プロセッサは、共有メモリに貯えられている S_1 の任意の点を一定時間で読み込める。

ステップ4.1: 文献[7, 8]の技法を用いて、 $CH(S_1)$ と $CH(S_2)$ (これらは既に $VOR(S_1)$, $VOR(S_2)$ を構成するときに作られている)を併合して、 $CH(S)$ を作る。この後、3.2節で述べた集合 T_1 , T_2 を抽出する。但し、 T_1 , T_2 それぞれの要素はY座標に従ってソートしておく。

今後、 S_1 の点を担当しているプロセッサ(認識番号 $1 \sim N/2$)について考える。 S_2 の点に対しても同様に考えればよい。

ステップ4.2: $VOR(S_1)$ の全てのVoronoi vertexの中で自分の担当するvertexを求める。

S_1 の点のVoronoi diagramは、高々 $2(N/2) - 5 = N - 5$ 個のvertexを持っているので、1つのプロセッサ当り6個のVoronoi vertexを担当すれば、 $VOR(S_1)$ を構成するすべてのVoronoi polygonのすべてのvertexが、 S_1 の点を担当している $N/2$ 個のプロセッサに割り当てられる。

割り当ての方法を簡単に示そう。 P_1 から担当している点までの全ての点のVoronoi polygonのvertexの累積数を求める。このようにすれば、各プロセッサは、認識番号*6の数値に従って自分が担当すべき6つのvertexの1番目がどの点のVoronoi polygonのvertexであるか2分探索で分かる。 S_1 の各点のVoronoi polygonのvertexはAVL木として貯えられているので、1番目に担当すべきvertexはAVL木のトップダウン(top down)方式の探索で分かる。1番目が分かれば他の5つは $O(1)$ で全て分かる。但し、 $VOR(S_2)$ のVoronoi vertexを担当することはない。

ステップ4.3: 各プロセッサは、担当している S_1 の点のchain δ を作成する。

chain $\delta(i)$ ($P_i \in S_1$)の作成について述べよう($\delta(i)$ の定義については3.2節を参照)。 $\delta(i)$ は、高々 $(N/2 - 1)$ 個のVoronoi vertexを連結して作成されるchainである。この全てのvertexを求めるのに $O(N)$ かかる。よって、ここでは、そのchainの中の高々 O

($\log N$)個のvertexだけを求める¹⁰。これはShort cut技法を用いて作成する。9個のvertexを連結して作成された $\delta(i)$ の例をFig. 4に示す。図で○印のvertexのみが実際には求められる。

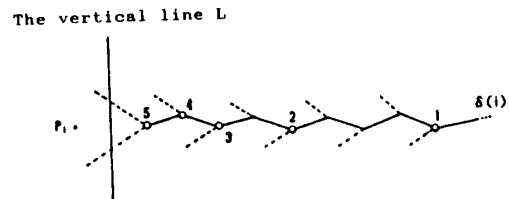


Fig. 4 chain $\delta(i)$.

ステップ4.4: 各プロセッサは担当しているvertexのうち垂直線Lより左側にあるvertexの各々が、 T_2 のどの点のVoronoi polygonの内部にあるか調べる。

これは、 T_2 を2分探索することによって実行される。 T_2 はY座標に従ってソートされており、その中間点(median)を P_m としよう。もし、担当しているvertex v が $V(P_m)$ の内部に存在すればそれで終了となる。さもなくて、もし、 v をX軸上に投影した点が $V(P_m)$ をX軸上に投影した区間内に存在するとき、 v が $V(P_m)$ より上にあれば P_m より上の T_2 の点の集合を、逆に、下にあれば P_m より下の T_2 の点の集合を新たに2分検索する。上記2つ以外るとき(必ず $P_m \in C_2$ である)は、ステップ4.3で作成されたchain $\delta(m)$ を探索し v が $\delta(m)$ より上にあるか下にあるかを調べる。このとき、もし、 v がchain $\delta(m)$ より上にあれば P_m より上の T_2 の点の集合を、逆に、下にあれば P_m より下の T_2 の点の集合を新たに2分検索する。 $\delta(m)$ の探索が $O(\log N)$ で実行できることは、文献[10]を参照されたい。

【命題1】 $VOR(S_1)$ の隣接している任意の2つのvertexを v_1, v_2 , edge $\overline{v_1 v_2}$ をそのVoronoi polygonのedgeとする2つの S_1 の点を P_{11}, P_{12} とする。

仮定 1. 1 円 $C(v_1)$, $C(v_2)$ の内部に S_2 の点が存在する。

仮定 1. 2 $X(P_{11}) < X(P_{12})$

仮定 1. 3 $X(v_1) < X(v_2)$

とする。このとき、次が言える。

(1) v_1 を通りX軸に平行な直線に対して、 P_{12} が v_2 と反対側に存在する。

→ 円 $C(v_1)$ の内部の S_2 の点は円 $C(v_2)$ の内部に含まれる。

(2) v_2 を通りX軸に平行な直線に対して、 P_{12} が v_1 と反対側に存在する。

→ 円 $C(V_2)$ の内部の S_2 の点は円 $C(v_1)$ の内部に含まれる。

(証明) 省略 (文献[10]参照)

ステップ 4.5: 各プロセッサは、担当している6つの vertex の各々が、 σ 作成に利用されるか否かを決める。

担当している vertex を v とする。

(1) v が半直線の Voronoi edge に隣接していない場合:

(1-1) v が垂直線 L 上あるいは L の左側にあるとき:

円 $C(v)$ の内部にステップ 4.4 で見つけた T_2 の点が存在しなければ、円 $C(v)$ の内部に S_2 の点は存在しない。したがって、この場合は、 v は $VOR(S)$ の Voronoi vertex として残され、 σ の作成のための vertex から除く。円 $C(v)$ の内部にステップ 4.4 で見つけた T_2 の点が存在する場合は、 v をその Voronoi polygon の vertex にもつ S_1 の点 P_1 を考える。 $V(P_1)$ を右回りに見て、 v が upper hull 上の点のときは次の vertex を、lower hull 上の点のときは前の vertex を見る。この vertex を v' とする。 v と v' が同じ hull 上の点の場合 (すなわち、 v が $V(P_1)$ の右端の点以外) のときはステップ(a)へ、さもなければ、ステップ(b)へ。

ステップ(a): v と v' がそれぞれ命題1の仮定および命題1の(1)の条件の v_1, v_2 を満たすか否かを調べる。もし、満たせば、 v を σ 作成のための vertex から除く (なぜなら、命題1の(1)より、円 $C(v)$ の内部の S_2 の点は円 $C(v')$ の内部に含まれるから、 v を利用して作成される σ の edge は v' を利用して作成されるため)。もし満たされなければ、ステップ(b)へ。

ステップ(b): v が upper hull 上の点のときは前の vertex を、lower hull 上の点のときは次の vertex を見る。この vertex を v'' とする。 v と v'' が同じ hull 上の点の場合 (すなわち、 v が $V(P_1)$ の左端の点以外) のときは、 v と v'' がそれぞれ命題1の仮定および命題1の(2)の条件の v_2, v_1 を満たすか否かを調べる。もし満たせば、 v を σ 作成のための vertex から除く (なぜなら、命題1の(2)より、円 $C(v)$ の内部の S_2 の点は円 $C(v'')$ の内部に含まれるから、 v を利用して作成される σ の edge は v'' を利用して作成されるため)。もし満たさないか、あるいは v と v'' が同じ hull 上の点ではないとき (すなわち v が $V(P_1)$ の左端の点のときは)、 v を σ 作成のための vertex として残す。

(1-2) v が垂直線 L の右側にあるとき:

v をその Voronoi polygon の vertex にもつ S_1 の点

P_1 を考える。 $V(P_1)$ を右回りに見て、 v が upper hull 上の点のときは次の vertex を、lower hull 上の点のときは前の vertex を見る。この vertex を v' とする。 v と v' が同じ hull 上の点の場合 (すなわち、 v が $V(P_1)$ の右端の点以外) のときは、円 $C(v)$ の内部の S_2 の点は円 $C(v')$ の内部に含まれるので、 v を σ 作成のための vertex から除く。 v と v' が同じ hull 上の点ではないとき (すなわち v が $V(P_1)$ の右端の点のときは)、 v を σ 作成のための vertex として残す。

(2) v が直線の Voronoi edge に隣接している場合:

(2-1) v が垂直線 L の左側にあるとき:

円 $C(v)$ の内部にステップ 4.4 で見つけた T_2 の点が存在しなければ、円 $C(v)$ の内部に S_2 の点は存在しない。したがって、この場合は、 v は $VOR(S)$ の Voronoi vertex として残され、 σ の作成のための vertex から除く。円 $C(v)$ の内部にステップ 4.4 で見つけた T_2 の点が存在する場合は、 v をその Voronoi polygon の vertex にもつ S_1 の点 P_1 を考える。 v が upper hull 上の点の場合を考える。 lower hull 上の点の場合も同様である。半直線の Voronoi edge の無限方向に仮想の vertex v^* を考える。3.2節で述べた半直線 H_1 は必ず σ と交わるので円 $C(v^*)$ の内部には S_2 の点が存在する (すなわち、vertex v^* は $VOR(S)$ における Voronoi vertex として残らないと考えてよい)。このとき、 v と v^* は命題1の仮定の v_1, v_2 を満足する。よって、 v と v^* が命題1の(1)または(2)の条件の v_1, v_2 を満たすか否かを調べて、もし、満たせば、 v を σ 作成のための vertex から除く (なぜなら、円 $C(v)$ の内部の S_2 の点は円 $C(v^*)$ の内部に含まれるから、 v を利用して作成される σ の edge は v^* を利用して作成されるため)。さもなければ、 v を σ 作成のための vertex として残す。また、どちらのときも、vertex v^* すなわち半直線 H_1 は σ 作成のために利用される。

(2-2) v が垂直線 L 上あるいは L の右側にあるとき:

v をその Voronoi polygon の vertex にもつ S_1 の点 P_1 を考える。 v が upper hull 上の点の場合を考える。 lower hull 上の点の場合も同様である。半直線の Voronoi edge の無限方向に仮想の vertex v^* を考える。このときは、円 $C(v)$ の内部に S_2 の点が存在すれば、その S_2 の点は円 $C(v^*)$ の内部の点でもあるので、 v を σ 作成のための vertex から除く。 vertex v^* すなわち半直線 H_1 は σ 作成のために利用される。

以上で、ステップ 4.5 を終わる。

$P_r \in S$ とする。 $VOR(S_1)$ [あるいは $VOR(S_2)$] における $V(P_r)$ と交わる σ の任意の subchain を σ_r とする。

σ_r によって分割された $V(P_r)$ の2つの部分多角形のうち点 P_r を含まない部分多角形を $V(P_r, \sigma_r)$ と記す。

【命題 2】 ステップ4. 5で残された vertex (あるいは半直線) を用いれば σ が作成できる。

(証明) $VOR(S_1)$ における $V(P_1)$ と交わる σ の任意の subchain を σ_1 とする。このとき部分多角形 $V(P_1, \sigma_1)$ の vertex のうち、ただ1つの vertex がステップ4. 5で残された vertex v となること、また、その他の vertex の円 C の内部に S_2 の点が含まれるならば、その S_2 の点は円 $C(v)$ の内部にも含まれることを言えばよい。これらはステップ4. 5の手順により容易に成り立つのが分かる。

$P_1 \in S_1, P_k \in S_2$ とし、 $v_1[v_k]$ を $VOR(S_1)[VOR(S_2)]$ における $P_1[P_k]$ の Voronoi polygon $V(P_1)[V(P_k)]$ の任意の vertex (あるいは半直線) とする。以後記述の簡単のため、半直線も vertex と呼ぶことにする。円 $C(v_1)$ の内部に点 P_k を含み、しかも円 $C(v_k)$ の内部に点 P_1 を含む場合は、 v_1 と v_k は、 σ 作成(P_1 と P_k の共通辺の作成)に利用可能な頂点对であるという。

次のステップ4. 6では、 σ 作成に利用可能な頂点对(これらの vertex としては、ステップ4. 5で残された vertex が利用される)を見つけ出し、これらの頂点对を用いて σ を作成する。すなわち、 v と v' がそれぞれステップ4. 5で残された $VOR(S_1)$ における Voronoi polygon $V(P_1)$ 、 $VOR(S_2)$ における Voronoi polygon $V(P_k)$ の vertex で、しかも v と v' が σ 作成に利用可能な頂点对であれば、 $V(P_1)$ と $V(P_k)$ と線分 $\overline{P_1 P_k}$ の垂直二等分線との intersection を求め、それを P_1 と P_k の共通辺とする。

ステップ4. 6を述べる前に2, 3の準備をする。

$P_1 \in O_1[P_k \in O_2]$ のとき半直線 $H_1[H_k]$ または $H'_1[H'_k]$ を、そして $P_1 \in C_1[P_k \in C_2]$ のとき最も右側(左側)の vertex を、共に $v_{1l}[v_{kl}]$ と記す。明らかに、 $v_{1l}[v_{kl}]$ は、ステップ4. 5で残された σ 作成のために利用される vertex または半直線である。

【命題 3】 S_1, S_2 の任意の点をそれぞれ P_1, P_k とし、 $v_1[v_k]$ をステップ4. 5で残された $V(P_1)[V(P_k)]$ の任意の vertex とする。 v_1 と v_k が σ 作成に利用可能な頂点对であるための必要条件は $v_1=v_{1l}$ であるかあるいは $v_k=v_{kl}$ であることである。

(証明) 省略(文献[10]参照)

【命題 4】 $P_1 \in S_1, P_k \in T$ (T は T_2 の任意の部分集合)で、 $V(P_1)$ の任意の vertex を v_1 とし、 v_1 と v_{kl} が σ 作成(P_1 と P_k の共通辺の作成)に利用可能な頂点对とする。また P_1 と P_k の共通辺を e 、 e の両端点で Y 座標の大きい方を g 、小さい方を s とする。また、 P_k

より上の T の点集合を T_g, P_k より下の T の点集合を T_s とする。このとき、

(1) 点 g が $V(P_1)$ の edge 上、点 s が $V(P_k)$ の edge 上にある。

→ T_g の任意の点 P_r の Voronoi polygon $V(P_r)$ の任意の vertex v_r に対し、 v_1 と v_r は σ 作成(P_1 と P_r の共通辺の作成)に利用可能な頂点对ではない。

(2) 点 s が $V(P_1)$ の edge 上、点 g が $V(P_k)$ の edge 上にある。

→ T_s の任意の点 P_u の Voronoi polygon $V(P_u)$ の任意の vertex v_u に対し、 v_1 と v_u は σ 作成(P_1 と P_u の共通辺の作成)に利用可能な頂点对ではない。

(3) 点 g と点 s の両方とも $V(P_1)$ の edge 上にある。

→ P_k 以外の T_2 の任意の点 P_z の Voronoi polygon $V(P_z)$ の任意の vertex v_z に対し、 v_1 と v_z は σ 作成(P_1 と P_z の共通辺の作成)に利用可能な頂点对ではない。

逆に、 $P_1 \in T$ (T は T_1 の任意の部分集合)、 $P_k \in S_2$ で $V(P_k)$ の任意の vertex についても同様である。

(証明) 省略(文献[10]参照)

【命題 5】 $P_1 \in S_1, P_k \in T$ (T は T_2 の任意の部分集合)で、 $V(P_1)$ と $V(P_k)$ と線分 $\overline{P_1 P_k}$ の垂直二等分線との intersection (P_1 と P_k の共通辺) I_{1k} ($\in \sigma$)が存在するとし、その Y 座標の大きい方の端点を g 、小さい方の端点を s とする。このとき、円 $C(v_x)$ の内部に S_2 の点が存在するような $VOR(S_1)$ の任意の vertex v_x について以下のことが成り立つ。但し、 v_x をその点の Voronoi polygon の vertex にもつ点を $P_x(\in S_1)$ とし、 P_k より上の T の点集合を T_g, P_k より下の T の点集合を T_s とする。

(1) vertex v_x が P_1 と P_k の共通辺 I_{1k} の作成に利用可能な頂点对の一方となる($P_1=P_x$)の場合。

→ v_1 を v_x で置き換えた命題4の(1), (2), (3)が成り立つ。

(2) (1)以外で、

$$Y(P_x) < Y(s), Y(v_x) < Y(g)$$

→ T_g の任意の点 P_r の Voronoi polygon $V(P_r)$ の任意の vertex v_r に対し、 v_x と v_r は σ 作成(P_x と P_r の共通辺の作成)に利用可能な頂点对ではない。

(3) (1), (2)以外で、

$$Y(P_x) > Y(g), Y(v_x) > Y(s)$$

→ T_s の任意の点 P_u の Voronoi polygon $V(P_u)$ の任意の vertex v_u に対し、 v_x と v_u は σ 作成(P_x と P_u の共通辺の作成)に利用可能な頂点对ではない。

(4) (1)~(3)以外で,

(a) P_x から v_x へ向かって, I_{ik} が左側に存在.
 → T_c の任意の点 P_r のVoronoi polygon $V(P_r)$ の任意のvertex v_r に対し, v_x と v_r は σ 作成 (P_x と P_r の共通辺の作成) に利用可能な頂点对ではない ((2)と同じ).

(b) P_x から v_x へ向かって, I_{ik} が右側に存在.
 → T_s の任意の点 P_u のVoronoi polygon $V(P_u)$ の任意のvertex v_u に対し, v_x と v_u は σ 作成 (P_x と P_u の共通辺の作成) に利用可能な頂点对ではない ((3)と同じ).

逆に, $P_i \in T$ (T は T_1 の任意の部分集合), $P_k \in S_2$ で, $VOR(S_2)$ の任意のvertex についても同様である.

(証明) 省略 (文献[10]参照)

【命題 6】 $P_i \in S_1$, $P_k \in T$ (T は T_2 の任意の部分集合)で, $V(P_i)$ と $V(P_k)$ と線分 $\overline{P_i P_k}$ の垂直二等分線とのintersection (P_i と P_k の共通辺) $I_{ik} (\in \sigma)$ が存在するとし, その Y 座標の大きい方の端点を g , 小さい方の端点を s とする. このとき, 円 $C(v_w)$ の内部に S_1 の点が存在するような $VOR(S_2)$ の任意のvertex を v_w , v_w をその点のVoronoi polygonのvertexにもつ点を $P_w (\in S_2)$ とする. このとき, v_w について以下のことが成り立つ.

(1) vertex v_w が P_i と P_k の共通辺 I_{ik} の作成に利用される頂点对の一方となる ($P_k = P_w$ の場合).

(a) 点 g が $V(P_w)$ のedge上, 点 s が $V(P_i)$ のedge上にある.

→ v_w と命題5の(3)と(4)(b)の条件を満たす v_x は σ 作成 (P_w と P_x の共通辺の作成) に利用可能な頂点对ではない.

(b) 点 s が $V(P_w)$ のedge上, 点 g が $V(P_i)$ のedge上にある.

→ v_w と命題5の(2)と(4)(a)の条件を満たす v_x は σ 作成 (P_w と P_x の共通辺の作成) に利用可能な頂点对ではない.

(c) 点 g と点 s の両方とも $V(P_w)$ のedge上にある.

→ P_i 以外の T_1 の任意の点 P_c のVoronoi polygon $V(P_c)$ の任意のvertex v_c に対し, v_w と v_c は σ 作成 (P_w と P_c の共通辺の作成) に利用可能な頂点对ではない.

(2) (1)以外で,

$$Y(P_w) < Y(s), Y(v_w) < Y(s)$$

→ v_w と命題5の(3)と(4)(b)の条件を満たす v_x は σ 作成 (P_w と P_x の共通辺の作成) に利用可能な頂点ではない.

(3) (1), (2)以外で,

$$Y(P_w) > Y(g), Y(v_w) > Y(g)$$

→ v_w と命題5の(2)と(4)(a)の条件を満たす v_x は σ 作成 (P_w と P_x の共通辺の作成) に利用可能な頂点对ではない.

(4) (1)~(3)以外で,

(a) P_w から v_w へ向かって, I_{ik} が右側に存在.

→ v_w と命題5の(3)と(4)(b)の条件を満たす v_x は σ 作成 (P_w と P_x の共通辺の作成) に利用可能な頂点对ではない. ((2)と同じ).

(b) P_w から v_w へ向かって, I_{ik} が左側に存在.

→ v_w と命題5の(2)と(4)(a)の条件を満たす v_x は σ 作成 (P_w と P_x の共通辺の作成) に利用可能な頂点对ではない ((3)と同じ).

(証明) 省略 (文献[10]参照)

ステップ 4.6 : σ を作成する.

σ を作成するステップ4.6は, 次の3つのサブステップからなる. 前述の集合 T_1, T_2 の各点は, Y 座標に従ってソートされているとする. T_2 の中点 P_m より上にある T_2 の点の集合 T_{2G} , P_m より下にある T_2 の点の集合を T_{2S} とする).

ステップ 4.6.1

ステップ4.5で残された $VOR(S_1)$ 中の各vertex v を担当しているプロセッサ $P(v)$ は, 次のことを実行する. いま, v は $V(P_i)$ ($P_i \in S_1$)のvertexであるとする.

$P(v)$ は, P_i と T_2 中の各点 P_k との共通辺 $I_{ik} (\in \sigma)$ を作成しようとする. $P(v)$ は, まず T_2 の中点 P_m を求め, P_i と P_m の共通辺 I_{im} が存在するか否かを調べる.

このことを調べるためには, 命題3より, v と v_{mt} (v_{mt} は, $P_m \in O_2$ のときは H_m または H'_m , $P_m \in C_2$ のときは $V(P_m)$ の最も左側のvertex)が P_i と P_m の共通辺の作成に利用可能な頂点对であるか否かを調べればよい.

もし, 利用可能な頂点对であるならば, P_i と P_m の垂直二等分線と $V(P_i)$ と $V(P_m)$ のintersectionを求め, それを $I_{im} (\in \sigma)$ とする. 次に T_2 の他の点と P_i との共通辺の作成に入る. この場合, P_i と T_2 の他のどの点とが

(σ に含まれる)共通辺を持つ可能性があるかどうかは, 前述の命題4~命題6を用いて知ることができる. すなわち, 命題4~6により, 次に P_i と共通辺を持つ可能性のある(P_m 以外の) T_2 の点は, T_{2G} か T_{2S} のいずれか一方のみに存在することを知ることができる. いま, 例えば, P_i との共通辺が存在する点が T_{2G} 中に存在する可能性があるかと判断された場合は, $P(v)$ は, 今度は

T_{2c} 中の中点 P_m を求め、上述と同様に P_i と $P_{m'}$ との共通辺 $I_{im'}$ の作成に入ろうとする。すなわち、 V と $V_{m'}$ が P_i と $P_{m'}$ の共通辺の作成に利用可能な頂点对であるか否かを調べ、もし、利用可能な頂点对であるならば、 P_i と $P_{m'}$ の垂直二等分線と $V(P_i)$ と $V(P_{m'})$ の intersection を求め、それを $I_{im'} (\in \sigma)$ とする。再び、命題 4～6 を用いて、次に P_i と共通辺を持つ可能性のある T_2 の点は、 T_{2c} の上半分の点であるか下半分の点であるかを定める。以下、同様にして $P(v)$ は、 P_i と T_2 の残りの点との共通辺を作成していく。結局、 $P(v)$ は、 P_i が T_2 のどの点と共通辺を持つかを定めるのに T_2 の 2 分探索を実行すればよいことに注意されたい。

ステップ 4.6.2

ステップ 4.5 で残された $VOR(S_2)$ 中の各 vertex v を担当しているプロセッサ $P(v)$ が、 P_k (但し、 P_k は $V(P_k)$ が v をその vertex としてもつような S_2 の点である) と T_1 の各点 P_i との共通辺 (もし存在すれば) I_{ik} を作成する。

この作成の方法は、ステップ 4.6.1 と同様である。

ステップ 4.6.3

ステップ 4.6.1, 4.6.2 で得られた各共通辺を整理統合して σ を作成し、 S 中の各点の ($VOR(S)$ における) Voronoi polygon を表す AVL 木を作成する。

4. 評 価

ここでは、ステップ 4 の実行時間を評価する。ステップ 4.1 は、文献[7, 8]の Convex hull の並列アルゴリズムより $O(\log N)$ で実行できる。ステップ 4.2 とステップ 4.3 は、明らかに $O(\log N)$ で実行できる。ステップ 4.4 が $O(\log^2 N)$ で実行できることを示そう。 S_1 の点を担当した任意のプロセッサについて考える。ステップ 4.4 の実行時間を $T_{4.4}(N)$ とすると、

$$T_{4.4}(N) = T_{4.4}(N/2) + cf(N) \quad c \text{ は定数}$$

となる。ここで、 $f(N)$ は vertex $v (\in VOR(S_1)) (\in VOR(S_2))$ が $T_2[T_1]$ のある点の Voronoi polygon の内部にあるか否か、もし、ないならば、上か下のどちらをさらに探索するべきか調べるのにかかる実行時間である。文献[10]より $f(N) = O(\log N)$ であることが示される。よって、 $T_{4.4}(N) = O(\log^2 N)$ となる。

ステップ 4.5 は、各プロセッサは明らかに $O(1)$

で実行できる。

ステップ 4.6, つまり σ 作成の実行時間を示す。サブステップ 4.6.1 の実行時間は $O(\log^2 N)$ である。

なぜなら、 T_2 の各探索において共通辺を作るのには $O(\log N)$ 時間で十分であり、 T_2 の 2 分探索は $O(\log N)$ 回で終了するからである。同様に、サブステップ 4.6.2 の実行時間は $O(\log^2 N)$ である。サブステップ 4.6.3 の実行時間は $O(\log N)$ であることが示される。従って、ステップ 4.6 の実行時間は $O(\log^2 N)$ である。詳細は文献[10]を参照されたい。

以上より、ステップ 4 の実行時間はステップ 4.4, 4.6 の実行時間に存在し、 $O(\log^2 N)$ となる。

5. 結 論

本研究では、並列計算機モデルを用いて幾何学上の問題を解く一例として、平面上の N 個の点の集合 S の Voronoi diagram $VOR(S)$ の並列アルゴリズムを与えた。

本稿の Voronoi diagram の並列アルゴリズムは $O(\log^3 N)$ である。平面上の N 個の点集合 S の Voronoi diagram $VOR(S)$ を N 個のプロセッサを用いて $O(\log N)$ で構成するための並列アルゴリズムの開発は、今後の研究課題である。

参 考 文 献

- 1) F. P. Preparata and M. I. Shamos: "Computational Geometry", Springer-Verlag, 1985.
- 2) Y. Shiloach and U. Vishkin: "Finding the maximum, merging, and sorting in a parallel computation model", J. of Algorithms, 2, pp. 88-102 (1981).
- 3) U. Vishkin: "An optimal parallel connectivity algorithms", Discrete Applied Mathematics, 9, pp. 197-207 (1984).
- 4) M. Ajtai, J. Komlos, W. L. Steiger and E. Szemerédi: "Deterministic selection in $O(\log \log n)$ parallel time", Proc. of the 18th ACM STOC, pp. 188-195 (1986).
- 5) M. Ajtai, J. Komlos and E. Szemerédi: "Sorting in $\log n$ parallel step", Combinatorica 3(1), pp. 1-19 (1983).
- 6) A. Aggarwal, B. Chazelle, L. Guibas, C. O'Dunlaing and C. Yap: "Parallel computational geometry", Proc. of the 26th IEEE FOCS, pp.

- 468-477 (1985) .
- 7) M. J. Atallah and M. T. Goodrich : "Efficient parallel solutions to geometric problems", *Parallel Processing, IEEE*, pp. 411-417 (1985) .
- 8) D. Nath, S. N. Maheshwar and P. C. P. Bhatt : "Parallel algorithms for the convex hull problem in two dimensions", *Conference On Analysis Problem Classes and Programming for Parallel Computing*, pp. 358-372 (1981) .
- 9) D. E. Knuth: "Sorting and Searching", *THE ART OF COMPUTER PROGRAMMING, Vol. 3*, pp. 111-114, 451-469 (1973) .
- 10) 阿部達朗, "計算幾何学における並列アルゴリズム", 昭和61年度山口大学工学部大学院修士論文 (1987) .

(昭和63年 3 月15日受理)