

制約付き無交差全域木列挙アルゴリズム

加藤 直樹* 谷川 眞一*

概要 平面上の n 点からなる点集合 P と P 上の無交差な辺の集合 F が与えられた際、 P 上の F を部分集合として含む全域木を F 制約付き全域木と呼ぶ。本論で提案するアルゴリズムは P 上の F 制約付き無交差全域木を一つ当たり $O(n^2)$ の計算時間で列挙を行う。

Computing All Non-crossing Spanning Trees on a Point Set Containing Required Edge Set

Naoki Katoh* Shin-ichi Tanigawa*

Abstract In this paper we present an algorithm for enumerating without repetitions all non-crossing geometric spanning trees on a given set of n points in the plane including a given necessary edge set. Our enumeration algorithm generates each output graph in $O(n^2)$ time and $O(n)$ space based on reverse search technique. This result improves the previous $O(n^3)$ bound for the unconstrained case by factor of $O(n)$. Our algorithm deal with the edge-constrained case in the same running time.

1 はじめに

頂点数 n の頂点集合 $V = \{1, \dots, n\}$ と辺数 m の辺集合 E から構成されるグラフ $G = (V, E)$ が連結かつサイクルを含まないとき、 G は全域木である。平面上の点集合 $P = \{p_1, \dots, p_n\} \subset \mathbf{R}^2$ が与えられた際、頂点 $i \in V$ を $p_i \in P$ へ、辺 $ij \in E$ を直線分 $p_i p_j$ へと埋め込んで得られるグラフを P 上の幾何グラフと呼ぶ。幾何グラフの2つの辺がそれらの端点以外で共有点を持たない時、グラフは無交差である。

本論では幾何全域木を単に全域木と呼ぶことにする。点集合 P の頂点を結ぶ無交差な直線分の集合 F が与えられた際、 F を部分集合として含む無交差全域木を F -制約付き無交差全域木と呼ぶ。本論では一般の点配置の P 上における全ての F -制約付き無交差全域木の列挙を行うアルゴリズムの提案する。本論では平面上に埋め込まれたグラフのみを想定することら、点 p_i を頂点 i 、直線分 $p_i p_j$ を単に辺 ij と記す。

我々が本論で提案を行う列挙アルゴリズムは、 $O(n)$ の記憶容量を用いて出力1つ当たり $O(n^2)$ の計算時間で F -制約付き無交差全域木の列挙を行う。この結果は辺制約なしの場合であっても、既往の手法として知られる Avis and Fukuda [6] による出力1つ当たり $O(n^3)$ 時間のアルゴリズムを計算時間において改善している。また我々の手法は、[6]での手法では拡張することができない、辺制約付き無交差全域木列挙を制約なしの場合と同じ計算時間で実現する。辺制約付き幾何グラフの列挙アルゴリズムとして、我々は最近、論文 [8] において、ドロネー三角形分割を用いた平面上の無交差ラーマンフレームワークの列挙アルゴリズムを提案した。そこにおいて、詳細な記述は行っていないが、ドロネー三角形を用いた手法と同様に平面上の全域木の列挙が $O(n^3)$ の計算時間で可能である事を記した。しかしこの手法では計算時間をこれ以上改善することができなかったため、今回異なる手法を考察し $O(n^2)$ の計算時間を実現した。

三角形分割 [6, 10, 12], 無交差全域木 [6], pseudo-triangulations [4, 9, 11], 無交差ラーマンフレームワーク [7, 8] 等、平面上の幾何グラフの列挙を行うアルゴリズムの研究は盛んに行われている。今、列挙を行うオブジェクト(幾何グラフ)の集合を O とした時、ある

* Department of Architecture and Architectural Engineering, Kyoto University, Katsura, Nishikyo-ku, Kyoto 615-8450 Japan, {naoki, is.tanigawa}@archi.kyoto-u.ac.jp. This research is supported by the project *New Horizons in Computing*, Grant-in-Aid for Scientific Research on Priority Areas, NEXT Japan.

オブジェクト $o_1 \in O$ から辺を一本取り除き新たに一本加えることによって新たなオブジェクト $o_2 \in O$ を得る局所探索操作を (1-) フリップと呼ぶ。2つのオブジェクトが一回のフリップによって相互に変換可能である時、それらは隣接しているものと捉え、各オブジェクトを頂点とし、隣接しているオブジェクトの頂点間を辺で結ぶことでグラフ \mathcal{G}_O が得られる。列挙アルゴリズムの基本的な考えはこの \mathcal{G}_O 上を効率よく辿ることにあるが、列挙アルゴリズムのみならず近傍探索に基づく最適化アルゴリズムやヒューリスティックの多くは \mathcal{G}_O 上の探索を行う事により実現されている。そのため \mathcal{G}_O の性質を解明することは各種アルゴリズムの有効性や性能を評価する際の重要な理論を与える。そのため、三角形分割 (e.g. [15]) や, pseudo-triangulations [1], 無交差辺マッチング [14], 単純多角形 [13], 無交差全域木 [1-3, 6] 等における \mathcal{G} , の連結性やその直径に関する研究は盛んである。特に我々の研究と関連が深いものとして, Avis and Fukuda [6] と Aichholzer ら [1-3] によるものが挙げられる。平面上の n 頂点上の無交差全域木の集合を ST と記した時, Avis and Fukuda [6] は $2n-4$ 回の 1-フリップで \mathcal{G}_{ST} が連結である事を示した。また 1-フリップではないフリップを用いて, Aichholzer ら [2] は $O(\log n)$ 回のフリップで \mathcal{G}_{ST} が連結である事を示し, また最近 [1] においてさらに改善を行った結果を得ている。Aichholzer らは [2, 3] において, *edge slide* と呼ばれる条件付 1-フリップの提案をしている。具体的に *edge slide* とは取り除く辺が一方の端点固定の状態では、もう一方の端点が全域木の辺に沿って移動し新たな辺へと移り変わる操作として定義がなされている。Aichholzer らは [3] において $O(n^2)$ 回の *edge slide* を用いて \mathcal{G}_{ST} が連結である事を示している。本論では、辺制約付き無交差全域木に対して、端点共有制約付きのフリップ、即ち取り除かれる辺と新たに加えられる辺が端点を共有している 1-フリップの提案を行い、全ての辺制約付き無交差全域木の集合が $O(n^2)$ 回の端点共有フリップと $O(n)$ 回の基底変換を用いて連結である事を新たに示す。

ここで本論で提案するアルゴリズムを支える手法として、Avis and Fukuda による逆探索手法と最小インデックス三角形分割について簡単に解説を行う。

逆探索手法 [5, 6] は連結な \mathcal{G}_O の全てのノードを少ないメモリで効率よく辿る手法である。そのために根と

呼ばれる特定のノードを定義し、さらに根以外の全てのノードに対して、(1) 唯一の親をもち (2) 自分自身が自分の祖先にならない、という条件を満たす親を定める。これにより探索木と呼ばれる \mathcal{G}_O 上の根付木が定まる。逆探索手法はこの探索木上を深さ優先探索の容量で効率よく全てのノードを辿りオブジェクトの列挙を行う。よって (根を唯一に定めた場合)、全てのオブジェクトの列挙を行うためには \mathcal{G}_O が連結であることを示さなければならない。

最小インデックス三角形分割のアイデアは、Bespamyatnikh [10] による三角形分割列挙のために考えられた辞書式順序最小三角形分割に由来する。我々は [10] で用いられた辞書式順序とは異なったインデックスを定義し、各三角形分割にインデックスを割り当てることで Bespamyatnikh による結果を辺制約付き三角形分割の場合へと一般化を行った。また辺制約付き三角形分割も辺制約なしの場合と同様に、 $O(n^2)$ 回のフリップ操作で各三角形分割が連結である事を示した。最小のインデックスを持つ三角形分割 (最小インデックス三角形分割) は我々の列挙アルゴリズムにおいて重要な役割を果たす。ここで提案する三角形分割を用いた列挙アルゴリズムの枠組みは、無交差全域木や無交差ラーマンフレームワークだけでなく他の無交差グラフ列挙アルゴリズムへの適用も期待できる。

2 最小インデックス三角形分割

本節において、Bespamyatnikh [10] による辞書式順序最小の三角形分割を辺制約付きの場合に拡張することで辺制約付き最小インデックス三角形分割の定義を行う。さらに任意の辺制約付き三角形分割が $O(n^2)$ 回のフリップ操作によって辺制約付き最小インデックス三角形分割に変換可能である事を新たに示す。ここで定義を行う辺制約付き最小インデックス三角形分割は、本論で提案する列挙アルゴリズムにおいて重要な役割を果たす。

2.1 諸定義

まず本論で用いられる幾つかの表記を定義しておく。 n 個の点から構成される頂点集合 P が与えられた際、 P の頂点を x 座標の順に $\{1, 2, \dots, n\}$ で表す。本論では P の全ての頂点は異なる x 座標を有し、全ての 3 頂点は同一直線上にはないものとする。各 2 頂点 $i, j \in P$ に対し、 i の x 座標が j より小さい時 $i < j$ と表す。各 $i \in P$

に対し、 $P_i = \{i+1, \dots, n\}$ と $P_i \subseteq P$ を定める。

本論では $i < j$ を満たす 2 点 i と j を結ぶ辺を ij と表す。頂点集合 P の 3 頂点 i, j, k に対して、三角形 Δijk の符号付面積 $\Delta(i, j, k)$ の符号を用いて k が i と j 上を通過する直線 l_{ij} の左に存在するか右に存在するかを判定することができる。即ち、 $\Delta(i, j, k) > 0$ ならば k は l_{ij} の左に存在し、 $\Delta(i, j, k) < 0$ ならば右に存在する。これを用いて辺の辞書式順序を次のように定める。辺 $e = ij, (i < j)$ と辺 $e' = kl, (k < l)$ に対して、 $i < k$ 又は、 $i = k$ かつ $\Delta(i, j, l) < 0$ が成立する時、 e は辞書式順序で e' より小さいとし、 $e < e'$ 又は $e' > e$ と記す。また、 e と e' が同じ辺を表している時、 $e = e'$ と記す。この定義は一般的に用いられている定義と異なるが、後の全域木の列挙を効率よく行うためにこの定義が必要である。また、 $i = k$ の時、我々の辞書式順序は i を中心とした時計回り順となっている事に注意しなければならない。

頂点集合 P の 2 頂点 i, j に対し、辺 ij が辺集合 F の辺と端点以外で交差しない時、 j は i から辺集合 F に関して可視であるという。頂点集合 P に対して、 i から可視な頂点の集合を $V_F(i, P)$ と表す。

頂点集合 P' に対して、 P' の凸包を $\text{conv}(P')$ と表す。頂点 $j \in P'$ と $i \notin \text{conv}(P')$ に対して、 $j = ij \cap \text{conv}(P')$ を満たすならば j は i から $\text{conv}(P')$ の境界辺に関して可視であると考え (図 1)。

辺 $e = ij$ に関して、その左端点及び右端点をそれぞれ $l(e), r(e)$ で表す。頂点 i と j 上を通過する直線は平面 \mathbf{R}^2 上の 2 つの開領域を定める。即ち、 $R^+(ij) = \{p \in \mathbf{R}^2 \mid \Delta(i, j, p) > 0\}$ 及び $R^-(ij) = \{p \in \mathbf{R}^2 \mid \Delta(i, j, p) < 0\}$ である。また同様に閉領域 $\bar{R}^+(ij), \bar{R}^-(ij)$ も定義することができる。また頂点 i 上を通過し x 軸に垂直な直線と考えた際、 $R^+(i) = \{p \in \mathbf{R}^2 \mid x(p) < x(i)\}$ 及び $R^-(i) = \{p \in \mathbf{R}^2 \mid x(i) < x(p)\}$ を定義することができる。但しここで $x(\cdot)$ は頂点の x 座標を表す。

頂点 $i \in P$ に対して、辺集合 F の要素でその左端点が i の辺集合を $F(i)$ と表す。頂点 i から $V_F(i, P_i)$ の凸包への 2 本の接線を考えた際、その接触点と i を結んだ辺をそれぞれ頂点 i の辺集合 F に対する上部接線及び下部接線と呼ぶ。但し辞書の小さい方を上部接線 ii^{up} 、もう一方を下部接線 ii^{low} と考える。この時、 $V_F(i, P_i) \subset \bar{R}^-(ii^{up})$ かつ $V_F(i, P_i) \subset \bar{R}^+(ii^{low})$ が成立していることに注意しなければならない。このことから

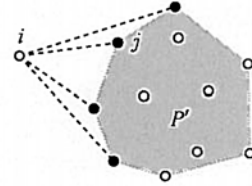


図 1 黒色頂点は i から凸包 $\text{conv}(P')$ に関して可視な頂点を表す。

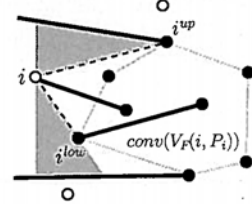


図 2 頂点 i の上部接線及び下部接線。黒色頂点は $V_F(i, P_i)$ を太線は F の辺を表す。影の領域が ii^{up} と ii^{low} の P に対する空領域となる。

ii^{up} と ii^{low} は以下に示すような P に対する空領域を定める。直線 l_i を i 上を通過し x 軸と垂直な直線とする。また $l_{ii^{up}}$ 及び $l_{ii^{low}}$ を点 i と ii^{up} 及び i と ii^{low} を通過する直線とする。この時、 l_i と交わる F の辺の内、 i の上部及び下部においてその交点が最も i に近い物をそれぞれ e_1 及び e_2 とする。すると l_i, e_1 (同様に e_2) 及び $l_{ii^{up}}$ (同様に $l_{ii^{low}}$) に囲まれた領域には P の要素は存在しない。もし e_1 又は e_2 が存在しない場合、この領域は $l_i, l_{ii^{up}}$ (同様に $l_{ii^{low}}$) に囲まれた部分となる。この性質を ii^{up} と ii^{low} の空領域条件と呼ぶ。

2.2 辺制約付き最小インデックス三角形分割

必要辺集合 F -制約付き最小インデックス三角形分割を以下のように定める。

定義 1 辺 ii^{up}, ii^{low} をそれぞれ $i \in P$ の上部接線及び下部接線とし、辺集合 $F(i) \cup \{ii^{up}, ii^{low}\}$ の各要素を i を中心とした時計周りの順に ii_0, ii_1, \dots, ii_k と表す。(よって $ii_0 = ii^{up}, ii_k = ii^{low}$ が成り立つ。) この時、 F -制約付き最小インデックス三角形分割が辺 $ij, (i < j)$ を含む必要十分条件は、 $0 \leq l \leq k-1$ を満たすある l において凸包 $H_l = \text{conv}(V_F(i, P_i) \cap \bar{R}^-(ii_l) \cap \bar{R}^+(ii_{l+1}))$ を考え、 j が i から H_l に関して可視であることである。(図 3)。

定義 1 の凸包 H_l を辺 ii_l と ii_{l+1} で挟まれた $V_F(i, P_i)$ の凸包と呼ぶ。図 4 に最小インデックス三角形分割の例

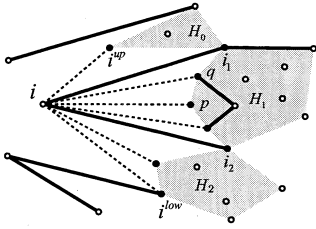


図3 頂点 i の周辺での最小インデックス三角形分割. ここで太線は F の辺を表す. 最小インデックス三角形分割は i と黒色頂点間を結ぶ辺をもつ.

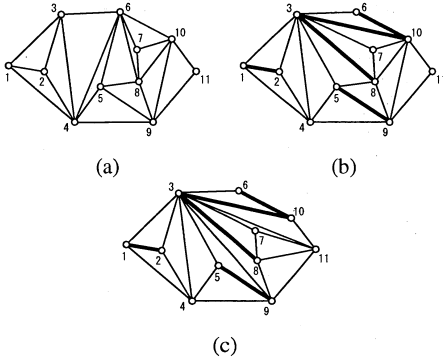


図4 (a) 最小インデックス三角形分割 ($F = \emptyset$). (b) 辺制約付き最小インデックス三角形分割. (c) 辺制約付き三角形分割.

を示す. 各頂点 i に対して, 最小インデックス三角形分割は必ず辺集合 $F(i) \cup \{ii^{up}, ii^{low}\}$ の要素を全て含むことに注意しなければならない.

頂点 i に関する帰納法を用いて以下を示すことは難しくない.

補題 1 F -制約付き最小インデックス三角形分割は頂点集合 P 上の三角形分割である.

2.3 辺制約付き三角形分割上のフリップ

本節において所与の頂点集合 P 上の F -制約付き最小インデックス三角形分割を T^* で表す. 頂点集合 P 上の任意の F -制約付き三角形分割 $T (T \neq T^*)$ に対してインデックスを定義する. インデックス, $index(T) = (n - c, d)$ は 2 つの整数, 臨界頂点のラベル c と臨界次数 d を用いる. ここで臨界頂点とは T^* に存在しないような辺に接続している頂点の中で最小の頂点, 臨界次数は臨界頂点に接続している T^* に含まれない辺の個数で

ある. また最小インデックス三角形分割のインデックスを $(0, 0)$ と定める. この時, $index(T) = (n - c, d)$ と $index(T') = (n - c', d')$ のインデックスを持つ 2 つの三角形分割 T と T' に対し, $c > c'$ 又は, $c = c'$ かつ $d < d'$ が成り立つ時, T のインデックスは T' のそれより小さいと考える. 臨界頂点のラベルが増加するに従い, 三角形分割のインデックスが減少することに注意しなければならない. 図 4(c) にインデックスが $(8, 2)$ の三角形分割を例示する.

三角形分割 T 上の辺 e に関して, T 上に e に接する 2 つの三角形が存在しかつそれらが凸四角形を形成する時, e はフリップ可能である. この時辺 e を取り除き, Q のもう一方の対角線を挿入する操作を (三角形分割の辺) フリップと呼び, 新たに得られた三角形分割のインデックスが前の三角形分割のそれより小さいならばそのフリップを改善フリップと呼ぶ. またその際, e は改善フリップ可能であると呼ばれる. 任意の辺制約付き三角形分割に対して以下の性質が成り立つ.

補題 2 最小インデックス三角形分割 T^* ではない任意の辺制約付き三角形分割 T において, T の臨界頂点を c とする. この時, c に接続し改善フリップ可能な辺が $T \setminus T^*$ 内に必ず存在する.

証明 三角形分割 T 上に必ず臨界頂点 c に接続する改善フリップが存在することを示そう. 辺集合 F 及び F -最小インデックス三角形分割 T^* の左端点が c と接続している辺の集合をそれぞれ $F(c)$, $T^*(c)$ と記す. また cc^{up} , cc^{low} を c の F 上における上部, 下部接線とする. ここで $T^*(c)$ の要素すべてが T に含まれていることを示す.

まず始めに cc^{up} が T に含まれていることを示す. ($cc^{low} \in T$ も同様に示すことができる.) 辺 cc^{up} が T に含まれていないと仮定すると, T は三角形分割である事から必ず cc^{up} と交差する辺 $e (e \neq T^*)$ が T に含まれている. よって, cc^{up} の空領域条件 (第 2.1 節) から, $l(e) < c$ が成り立つ. このことは頂点 $l(e)$ に辺 $e (e \neq T^*)$ が接続していることを示しており, c が臨界頂点であることに矛盾する.

次に $F(c) \cup \{cc^{up}, cc^{low}\}$ 以外の $T^*(c)$ の辺 cv が T に含まれていることを示そう. $F(c) \cup \{cc^{up}, cc^{low}\}$ の辺を c 周りの時計回り順で cc_0, \dots, cc_k と記す. 辺 cv は $T^*(c)$ に含まれていることから, 定義 1 より点 v が境界上に

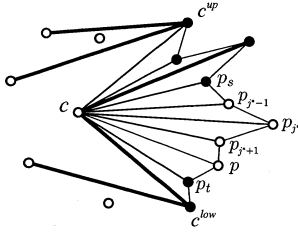


図5 改善フリップ cp_j の例。ここで太線は F を、黒色頂点は T^* 上で c に接続している辺を表す。

存在する辺 cc_l と cc_{l+1} で挟まれた $V_F(c, P_c)$ の凸包, H_l が必ず存在する。よって辺 cc_l , cc_{l+1} と点 c_l から c_{l+1} まで H_l の境界を辿ることで得られる凸な鎖によって1つの pseudo-triangle が定まる。また点 v はこの凸な鎖上に存在し、この pseudo-triangle 内に P の頂点は存在しない。辺 cv が T に存在しないと仮定すると、 T 内に cv と交差する辺 e が存在する。しかし辺 e は cv と交差することから、 e は少なくとも cc_l か cc_{l+1} のいずれかと交差し、 T が $F(c) \cup \{cc^{up}, cc^{low}\}$ の辺を全て持つことに矛盾する。よって $T^*(c) \subset T$ が成り立つ。

では T 上に c に接続する改善フリップが存在することを示そう。頂点 c は臨界頂点なので T^* に含まれない辺 cp が必ず存在する。よって辺集合 $T^*(c) \subset T$ の中で辞書順に連続する2つの辺で点 p を挟む辺のペア cp_s, cp_t が存在する(図5)。 T 上において c に接続する辺で cp_s と cp_t に挟まれるものを考え、そのような辺を辞書順に $cp_{s+1}, cp_{s+2}, \dots, cp_{t-1}$ と表す。 F -制約付き最小インデックス三角形分割上 T^* に三角形 $\Delta cp_s p_t$ が存在することから、全ての $j = s+1, \dots, t-1$ に対して $cp_j \in T \setminus T^*$ 及び $\Delta cp_s p_t$ 内に P の頂点が存在しない事が言える。このことから全ての p_j は辺 $p_s p_t$ と交差する。全ての p_j の内で点 p_s と p_t 上を通過する直線から(直線上の最近点までのユークリッド距離が)最も遠い点を p_j とした時、同一直線上に3つの頂点が存在しない事から、四角形 $cp_{j-1} p_j p_{j+1}$ は凸である。よって、 $c < p_{j-1}$ かつ $c < p_{j+1}$ が成り立っていることから、 cp_j をフリップさせることでよりインデックスの小さい三角形分割を得、 cp_j は改善フリップ可能である。 ■

補題2から以下の定理を得る。

定理1 全ての F -制約付き三角形分割 T は $O(n^2)$ 回の

フリップで F -制約付き最小インデックス三角形分割 T^* へ変換可能である。

証明 補題2より、 T^* でない T は必ず改善フリップ可能な辺を有し、この辺をフリップさせていくことで最小インデックスを有する T^* へ変換することができる。異なるインデックスの個数は $O(n^2)$ 個である事から定理を得る。 ■

3 辺制約付き無交差全域木

所与の頂点集合 P 上に、サイクルを持たない無交差な辺集合 F が与えられている。本節では P 上の F -制約付き無交差全域木の集合が $O(n^2)$ の(辺)フリップを用いて連結である事を示す。ここで P 上の F -制約付き無交差全域木の集合を ST と表す。

2つの辞書式順序で並べられた辺リスト $E = \{e_1 < e_2 < \dots < e_m\}$, $E' = \{e'_1 < e'_2 < \dots < e'_m\}$ に対して、 $e_i \neq e'_i$ が成立する最小の添え字 i で $e_i < e'_i$ が成り立つ時、 E は時書式順序で E' より小さい。

以後、辺制約 F に対する F -制約付き最小インデックス三角形分割を $T^*(F)$ と表すことにする。この時、 $T^*(F)$ に含まれる F -制約付き全域木を F -制約付き最小インデックス全域木と定め、その集合を $CSIST$ と記す。さらに $CSIST$ の中で辞書式順序最小の辺リストから構成される無交差全域木を ST^* と記す。

集合 $CSIST$ の各要素は、 $T^*(F)$ の辺集合を台集合とする閉路マトロイドの基である事から以下の補題が成り立つ。

補題3 集合 $CSIST$ のすべての F -制約付き無交差全域木の集合は $n-1$ 回のフリップで ST^* へ変換することができる。

では、 $CSIST$ 以外の F -制約付き無交差全域木 ST に対して、 ST が $CSIST$ のいずれかの要素とどれだけ異なっているかを表すために各 ST に対するインデックスの定義を行う。前節において各 F -制約付き三角形分割 T に対して、 T が $T^*(F)$ とどれだけ異なっているかを表すためにインデックス $index_F(T) = (n-c, d)$ を定めた(第2.3節)。我々は、各 F -制約付き無交差全域木 ST に対し、 ST -制約付き最小インデックス三角形分割を対応させる。つまり、 ST のインデックス、 $index(ST) = (n-c_{ST}, d_S T)$ を $index(ST) = index_F(T^*(ST))$ と定義する。また三角形分割の場合と同様に、 $c_{ST}(=c)$ を ST

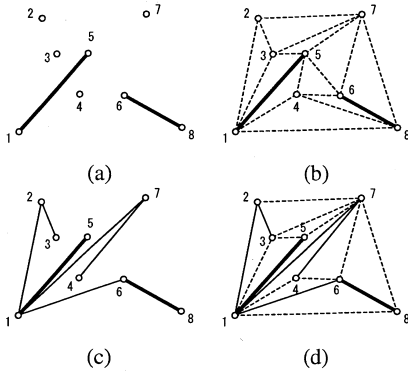


図6 (a) F , (b) $T^*(F)$, (c) ST , and (d) $T^*(ST)$,
ここで太線は F を点線は $T^*(ST) \setminus ST$ を表す.

の臨界頂点と呼ぶ. 例として, 図6の ST において臨界頂点は1, $\text{index}(ST)$ は $(7, 2)$ となる.

頂点集合 P の頂点 i について, $ST(i)$, $T^*(ST; i)$ を用いてそれぞれ ST 及び $T^*(ST)$ の辺で左端点が i である辺の集合を表すこととする. 無交差全域木 ST から ST -制約付き最小インデックス三角形分割を得る操作を考えた際, 三角形分割のために新たに追加される辺は定義1から明らかにフリップ不可能である. このことから以下を得る.

観察1 頂点 $i \in P$ に対して, $T^*(ST; i) \setminus (ST(i) \cup \{ii^{up}, ii^{low}\})$ 内のあらゆる辺は $T^*(ST)$ 上においてフリップ不可能である. 但しここで, ii^{up} 及び ii^{low} は i の ST 上における上部, 下部接線である.

補題2及び観察1を用いて以下を示すことができる.

補題4 F -制約付き無交差木 $ST \notin CSIST$ に対して, ST の臨界頂点を c とする. この時, (i) $T^*(ST) \setminus T^*(F)$ 内に少なくとも1つ改善フリップが存在し, (ii) 辺 $e \in T^*(ST)$ が改善フリップであるための必要十分条件は e がフリップ可能で $e \in ST(c) \setminus \{cc^{up}, cc^{low}\}$ が満たすことである. 但し cc^{up} , cc^{low} は c の ST 上での上部, 下部接線である.

補題5 すべての F -制約付き無交差全域木 ST は $O(n^2)$ 回のフリップで $CSIST$ の要素へ変換可能である.

証明 ST の臨界頂点を c とする. 補題4より $ST \setminus F$ 内に c に接続している改善フリップ可能な辺 cc^* が存在する. cc^* はフリップ可能であることから $T^*(ST)$ 上で

辺 cc^* に接する2つの三角形 $\Delta cc^*c_1^*$ と $\Delta cc^*c_2^*$ が存在する. 辺 cc^* を ST から取り除いた際, グラフ $G - cc^*$ は2つの連結成分に分割され, 頂点 c と c^* は異なる連結成分に属する. また頂点 c_1^* はどちらか一方にのみ属することから, cc_1^* または $c_1^*c^*$ を $ST - cc^*$ に加えることで新たな全域木 ST' を得る. また cc_1^* , $c_1^*c^*$ 及び ST の辺いづれも $T^*(ST)$ の辺である事から, 新たに得られる ST' は必ず無交差である. さらに ST' のインデックスは ST のそれより必ず小さくなっている. なぜなら取り除かれる辺 cc^* は $T(ST)$ 上において改善フリップ可能であり, cc^* の辺拘束を取り除くことによって三角形分割 $T^*(ST)$ 上で cc^* のフリップが必ず起きる. よって c に接続する $T^*(F)$ に含まれない辺の本数が必ず減少, つまり臨界次数は必ず減少する. 定理1から, この操作を $O(n^2)$ 繰返すことにより, 対応する三角形分割は $T^*(F)$ に至り, 結局全域木は $CSIST$ の要素へと変換される. ■

補題3と補題5より以下の定理が導かれる.

定理2 全ての F 制約付き無交差全域木は $O(n^2)$ 回のフリップを用いて連結である.

4 辺制約付無交差全域木の列挙

前節と同様に $CSIST$ の中で辞書式順序最小の辺リストから構成される無交差全域木を ST^* と記す. また F -制約付き無交差全域 ST の改善フリップの集合を I_{ST} と表す. 前節までの結果に基づき F -制約付無交差全域木の親を以下のように定める.

定義2 臨界頂点 c をもつ ST^* と異なる F -制約付き無交差全域木 ST において, F -制約付き無交差全域木 $ST' = ST - e_1 + e_2$ は辺 e_1, e_2 が以下を満たす時 ST の親である,

ケース1: $ST \in CSIST$,

• $e_1 = \max\{e \mid e \in ST \setminus ST^*\}$ かつ $e_2 = \min\{e \in ST^* \setminus ST \mid ST - e_1 + e \in ST\}$,

ケース2: $ST \notin CSIST$,

• $e_1 = cc^* = \min\{e \mid e \in I_{ST}\}$ かつ e_2 は $ST - e_1 + e_2 \in ST$ を満たす cc_1^* , $c_1^*c^*$ のどちらか一方, 但し頂点 c_1^* は $\Delta(c, c^*, c_1^*) > 0$ を満たす $T^*(ST)$ に含まれる三角形 $\Delta cc^*c_1^*$ の頂点である.

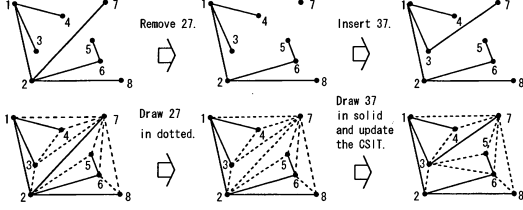


図7 無交差全域木 $ST \notin CSIST$ の親 ST' への移動. ここで $index(ST) = (6, 2)$, $index(ST') = (6, 1)$ である.

補題4より, I_{ST} は空集合でなくかつ $ST(c) \setminus F$ の部分集合である事に注意しなければならない. このことから定義2のケース2における e_1 は必ず存在する. また $T^*(ST)$ 上には cc^* に接する2つの三角形分割 $\Delta cc^*c_i^*$, $\Delta cc^*c_j^*$ が存在するが, 親を唯一に定めるためにここでは c_i^* を採用した. 図7に子から親への移動の様子を示す. 補題3と補題5から, 定義2によって第1節で述べた \mathcal{G}_{SP} 上の探索木が定められ事がわかる. ケース1, ケース2それぞれの場合に対して, 子から親への写像を $f_1: CSIST \setminus \{ST^*\} \rightarrow CSIST$, $f_2: ST \setminus CSIST \rightarrow ST$ と記す.

辺集合 ST' 及び K_n を辞書式順に並べた辺リストを $elist_{ST'}$, $elist_{K_n}$ とし, それぞれの第 i 番目の要素を $elist_{ST'}(i)$, $elist_{K_n}(i)$ とする. Avis and Fukuda [5, 6] の結果に基づき, 図8に我々の逆探索アルゴリズムを記す. 三角形分割 $T^*(ST)$ を求めるのに必要な計算時間を T_{CSIT} とした時, 子から親を取り出す作業には $O(n + T_{CSIT})$ の計算時間が必要である. よってこのままの単純な方法では第4行目から17行目までのwhileループに $O(n^3(n + T_{CSIT}))$ の計算時間が必要となる. 我々は第6行目から16行目までのループに必要な計算時間が $O(n)$ となる方法を考え, 第4行目から17行目までに必要な作業, つまり新たな子を探すのに必要な計算時間が $O(n^2)$ となる手法の開発を行った.

2つの異なる F -制約付き無交差全域木 ST, ST' において, $SP = ST' - e_{rem} + e_{add}$ となる辺 $e_{rem} \in ST' \setminus F$ 及び辺 $e_{add} \in K_n \setminus ST'$ が存在する場合を考える. 上述したアルゴリズムの効率化を図るためには ST が ST' の子であるかを効率よく確かめることが重要となる. よって $f_1(ST) = ST'$ または $f_2(ST) = ST'$ が成り立つ e_{rem} と e_{add} のペアの特徴付けを行い, アルゴリズムの効率化を図ることを考える.

Algorithm Enumerating F -constrained non-crossing geometric spanning trees.

```

1:  $ST^* := F$ -CSIST with lexicographically smallest
   edge list;
2:  $ST' := ST^*$ ;  $i, j := 0$ ;  $Output(ST')$ ;
3: repeat
4:   while  $i \leq |ST'|$  do
5:      $i := i + 1$ ;
6:     while  $j \leq |K_n|$  do
7:        $j := j + 1$ ;
8:        $e_{rem} = elist_{ST'}(i)$ ;  $e_{add} = elist_{K_n}(j)$ ;
9:       if  $ST' - e_{rem} + e_{add} \in ST$  then
10:         $ST := ST' - e_{rem} + e_{add}$ ;
11:        if  $f_1(ST) = ST'$  or  $f_2(ST) = ST'$  then
12:          $ST' := ST$ ;  $i, j := 0$ ;  $Output(ST')$ ;
13:         go to line 4;
14:        end if
15:       end if
16:     end while
17:   end while
18:   if  $ST' \neq ST^*$  then
19:     $ST := ST'$ ;
20:    if  $ST \in CSIST$  then  $ST' := f_1(ST)$ ;
21:    else  $ST' := f_2(ST)$ ;
22:    determine integers  $i$  and  $j$  such that  $ST' -$ 
       $elist_{ST'}(i) + elist_{K_n}(j) = ST$ ;
23:     $i := i - 1$ ;
24:   end if
25: until  $ST' = ST^*$ ,  $i = |ST'|$  and  $j = |K_n|$ ;

```

図8 F -制約付き無交差全域木列挙アルゴリズム.

以下に示す補題6及び補題7はそれぞれ e_{rem} と e_{add} のペアに対して $f_1(ST) = ST'$, $f_2(ST) = ST'$ が成り立つ必要十分条件を与える.

補題6 異なる $CSIST$ の要素 ST, ST' において, 辺 $e_{rem} \in ST' \setminus F$ 及び辺 $e_{add} \in K_n \setminus ST'$ を用いて $ST = ST' - e_{rem} + e_{add}$ が成立している時, $f_1(ST) = ST'$ が成り立つ為の必要十分条件は e_{rem}, e_{add} が以下の条件を満たすことである,

- (A) $e_{rem} \in ST^*$,
- (B) $e_{add} \in T^*(F) \setminus (ST^* \cup ST')$,

- (C) $e_{\text{rem}} < \min\{e \in ST^* \setminus ST' \mid ST' - e_{\text{rem}} + e \in ST\}$,
 (D) $e_{\text{add}} > \max\{e \mid e \in ST' \setminus ST^*\}$.

補題 7 異なる ST の要素 ST, ST' において, 辺 $e_{\text{rem}} = ab \in ST' \setminus F$ 及び辺 $e_{\text{add}} \in K_n \setminus ST'$ を用いて $ST = ST' - e_{\text{rem}} + e_{\text{add}}$ が成立している時, $f_2(ST) = ST'$ が成り立つ為の必要十分条件は $e_{\text{rem}}, e_{\text{add}}$ が以下の条件を満たすことである,

- (A) $e_{\text{add}} = \min\{e \mid e \in I_{ST}\}$,
 (B) $l(e_{\text{add}}) \leq a$, (i.e. $l(e_{\text{add}}) \leq l(e_{\text{rem}})$),
 (C) e_{rem} と e_{add} が端点を共有し, e_{add} が e_{rem} と点 a を共有する場合 $a \in R^+(e_{\text{add}})$ が, e_{add} が点 b を共有する場合 $b \in R^+(e_{\text{add}})$ が成り立つ,
 (D) 三角形 Δbx の面が $T^*(ST)$ に存在する.

提案するアルゴリズムは, 取り除く辺 e_{rem} を固定した状態で, 補題 6 又は補題 7 の必要十分条件を満たす辺 e_{add} の集合の列挙を行う. 我々はそれぞれの場合 (ケース 1, ケース 2) に対してこのような e_{add} の列挙を行う $O(n)$ 時間アルゴリズムの開発を行った. よって各 e_{rem} に対して, $ST' - e_{\text{rem}} + e_{\text{add}}$ が ST' の子となる辺のペア $(e_{\text{rem}}, e_{\text{add}})$ の集合を線形時間で列挙することができる. 可能な e_{rem} の個数は $O(n)$ である事から以下の定理を得る.

定理 3 F -制約付き無交差全域木の集合は $O(n)$ の記憶容量を用いてグラフ 1 つ当たり $O(n^2)$ の計算時間で列挙することができる.

紙面の都合上詳細は省略するが, e_{rem} 固定で各 e_{add} に対して上述した補題 7 の条件を直接確かめる方法では定理 3 の計算時間を実現する事は難しい. そのため補題 7 の各条件が成立する辺 e_{add} についてさらに細かい議論を行う必要がある. この先の議論は複雑になるのだが, 実際に実装を行う際障害となるような高度な手法は用いずに $O(n^2)$ アルゴリズムを実現している事を強調しておきたい.

参考文献

[1] O. Aichholzer, F. Aurenhammer, C. Huemer and H. Krasser. Transforming spanning trees and pseudo-triangulations. *Inf. Process. Lett.*, 97(1):19–22, 2006.
 [2] O. Aichholzer, F. Aurenhammer and F. Hurtado. Sequences of spanning trees and a fixed tree theorem. *Comput. Geom.*, 21(1-2):3–20, 2002.

[3] O. Aichholzer and K. Reinhardt. A quadratic distance bound on sliding between crossing-free spanning trees. In *Proc. EWCG04*, pages 13–16, 2004.
 [4] O. Aichholzer, G. Rote, B. Speckmann and I. Streinu. The zig-zag path of a pseudo-triangulation. In *Proc. WADS03*, LNCS 2748, pages 377–388, Ottawa, Canada, 2003. Springer Verlag.
 [5] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8:295–313, 1992.
 [6] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Appl. Math.*, 65(1-3):21–46, March 1996.
 [7] D. Avis, N. Katoh, M. Ohsaki, I. Streinu, and S. Tanigawa. Enumerating non-crossing minimally rigid frameworks. In *Proc. COCOON 2006*, LNCS 4112, pages 2005–215, 2006.
 [8] D. Avis, N. Katoh, M. Ohsaki, I. Streinu, and S. Tanigawa. Enumerating constrained non-crossing minimally rigid frameworks. http://arxiv.org/PS_cache/math/pdf/0608/0608102.pdf
 [9] S. Bereg. Enumerating pseudo-triangulations in the plane. *Comput. Geom. Theory Appl.*, 30(3):207–222, 2005.
 [10] S. Bespamyatnikh. An efficient algorithm for enumeration of triangulations. *Comput. Geom. Theory Appl.*, 23(3):271–279, 2002.
 [11] H. Brönnimann, L. Kettner, M. Pocchiola and J. Snoeyink. Enumerating and counting pseudo-triangulations with the greedy flip algorithm. In *Proc. ALENEX*, Vancouver, Canada, 2005.
 [12] A. Dumitrescu, B. Gärtner, S. Pedroni and E. Welzl. Enumerating triangulation paths. *Computational Geometry: Theory and Applications*, 20(1-2):3–12, 2001.
 [13] M. C. Hernando, M. E. Houle and F. Hurtado. On local transformation of polygons with visibility properties. In *Proc. COCOON00*, LNCS 1858, pages 54–63. Springer, 2000.
 [14] C. Hernando, F. Hurtado and M. Noy. Graphs of non-crossing perfect matchings., *Graphs and Combinatorics*, 18(3):517–532, 2002.
 [15] F. Hurtado, M. Noy and J. Urrutia. Flipping edges in triangulations. *Discrete & Computational Geometry*, 22(3):333–346, 1999.