社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－AL－108 （5）
2006／9／27

制約付き非交差静定グラフ列挙アルゴリズム

Davis Avis †　加藤 直樹 ‡　大崎 純 ‡　Ileana Streinu §　谷川 眞一 ‡

概要　本論では，平面上に一般の位置に与えられた n 頂点からなる頂点集合 P に対して，P 上に埋め込まれた，辺制約付き非交差静定グラフをすべて列挙するアルゴリズムについて述べる．静定グラフはラーマングラフとも呼ばれ，そのようなグラフ全体がマトロイドの基を作ることが知られている．一方，平面上に埋め込まれた無交差ラーマングラフ全体はマトロイドとはならない．本論文では，Avis and Fukuda による逆探索法を用いて，無交差ラーマングラフの列挙が可能であることを示す．またあらかじめ用いる辺が一部指定されている場合も列挙が可能である．

Enumerating Constrained Non-crossing Minimally Rigid Frameworks

David Avis †　Naoki Katoh ‡　Makoto Ohsaki ‡　Ileana Streinu §　Shin-ichi Tanigawa ‡

Abstract　In this paper we present and algorithm for enumerating without repetitions all the non-crossing generically minimally rigid bar-and-joint frameworks under edge constraint (also called constrained non-crossing Laman frameworks) on a given generic set of n points. Our algorithm is based on the reverse search paradigm of Avis and Fukuda. It generates each output graph in $O(n^3)$ time and $O(n^2)$ space.

1　はじめに

組合せ剛性理論 (combinatorial rigidity theory) とは，平面や空間内に伸び縮みのない棒 (straight bar) とジョイント (joint) を用いて組み立てられたフレームワーク (framework) の剛性，安定性を組合せ的に特徴付ける理論である．フレームワークの安定性に関する研究は多数の工学分野の研究者によって行われてきたが，近年，ロボットアームやリンク機構の設計，立体抽出等，その応用の多様さからますます注目を集めることとなった．フレームワークの安定性には，そのグラフ構造が大きく関わってきている事が知られるようになり，Laman [10] は，最小の棒から成る安定したフレームワークのグラフ構造に着目し，**ラーマングラフ**を提案した．頂点数 n，辺数 m のグラフ G について，$m = 2n - 3$ でかつ任意の n' 個の頂点部分集合においてその誘導部分グラフの辺数が $2n' - 3$ 以下である時，G をラーマングラフという．一般の点配置である頂点集合 P 上に埋め込まれたラーマングラフは**ラーマンフレームワーク**と呼ばれ，その集合は辺集合を台集合をとするマトロイドの基を構成するなど，優れた組合せ的特徴を有する事がわかっている．P 上の無交差な辺の集合を F とし，F を部分集合として含むラーマンフレームワークを F-制約付きラーマンフレームワークという．本論では，平面上に与えられた P 及び F に対する，F-制約付きラーマンフレームワークの列挙アルゴリズムを提案する．

本論で提案するアルゴリズムは，$O(n^2)$ の記憶容量を用いて，オブジェクト当たり $O(n^3)$ の計算時間で F-制約付きラーマンフレームワークの列挙を行う．文献 [4] において我々は辺制約なしの無交差ラーマンフレームワークの列挙アルゴリズムの提案を行った．そこでは自由度 1 のメカニズムの性質を用いたが，辺制約付きの列挙には拡張することができなかった．本論で提案するアルゴリズムは，剛性マトロイドの性質を用い，[4] とは異なる方法で，辺制約条件下でも同じ計算時間

† School of Computer Science, McGill University, Canada.
‡ Department of Architecture and Architectural Engineering, Kyoto University Katsura, Nishikyo-ku, Kyoto 615-8450 Japan, {ohsaki,naoki,is.tanigawa}@archi.kyoto-u.ac.jp.
Supported by JSPS Grant-in-Aid for Scientific Research on priority areas of New Horizons in Computing.
§ Dept. of Comp. Science, Smith College, Northampton, MA 01063, USA, streinu@cs.smith.edu. Supported by NSF grant CCF-0430990 and NSF-DARPA CARGO CCR-0310661.

と記憶容量で列挙が可能である．

ここで，ラーマンフレームワークに関係の深い幾何グラフである三角形分割及び pointed pseudo-triangulation (以下，PPT) の列挙問題と，無交差ラーマンフレームワークの列挙との違いについて述べておきたい．幾何グラフの列挙問題は，例えば無交差全域木 [3]，三角形分割 [3,7,9]，PPT [1,5,8] の列挙等，多数の問題に対して効率的なアルゴリズムの構築がなされている. 特に我々の手法と関連が深いのは Avis and Fukuda による逆探索手法 [2,3] を用いた Bereg [5,7] による三角形分割及び PPT の列挙アルゴリズムが挙げられる．

三角形分割や PPT の辺フリップと同様に，各無交差ラーマンフレームワークに対するフリップ操作を以下のように定める．

定義 1 無交差ラーマンフレームワーク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無交差ラーマンフレームワークを得る操作を ***L*-フリップ** (L-flip) と呼ぶ.

逆探索手法において 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**という．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同士を辺で結ぶことでグラフ $\mathcal{G}$ が得られる．逆探索手法はこの $\mathcal{G}$ 上を深さ優先探索の要領で辿ることで全てのオブジェクトの探索を行う．またアルゴリズムが辿る $\mathcal{G}$ 上の根付木のことを探索木 (search tree) と呼ぶ．よって逆探索手法を用いて全てのオブジェクトを列挙するためには，$\mathcal{G}$ が連結である事を示さなければならない．

三角形分割の場合，$O(n^2)$ 回の辺フリップで任意の三角形分割への変換が可能である事が知られており (第 3 節，定理 5)，このことから $\mathcal{G}$ は連結である．また PPT に関しては，Rote, Streinu and Santos [12] によって，$\mathcal{G}$ の $\mathbb{R}^{2n-3}$ 上の凸多面体への埋め込みが可能であることが示されている．このため $\mathcal{G}$ は連結であり，逆探索手法を用いて列挙が可能であることが知られている．一方，ラーマンフレームワークの列挙に関しては，任意の頂点集合上の (無交差とは限らない) ラーマンフレームワークの集合は，剛性マトロイドの基であることから，フリップ (基底変換) を用いてラーマンフレームワークの集合は連結であり，逆探索手法での列挙が可能である．しかしその部分集合である無交差ラーマンフレームワークの集合は必ずしも L-フリップを用いて連結であるとは限らない．本論において我々は，以下の定理が成り立つことを新たに証明する．

定理 1 全ての辺制約付きラーマンフレームワークは L-フリップを用いて連結である．

この定理に基づき, 以下の計算時間からなる辺制約付きラーマンフレームワーク列挙アルゴリズムを提案する．

定理 2 与えられた頂点集合上の辺制約付き無交差ラーマンフレームワークの集合を $O(n^2)$ の記憶容量を用いてオブジェクト 1 つあたり $O(n^3)$ の計算時間で求めることができる．

第 2 節では，この剛性理論に関する基本知識を，第 3 節では，我々のアルゴリズムに必要となる辺制約付き平面三角形分割の解説を行う．第 4 節で定理 1 の証明を，第 5 節で定理 2 の証明を与える．

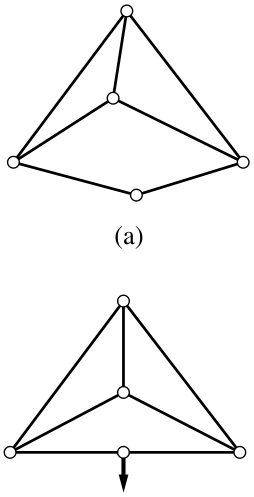
2 剛性理論

グラフ $G = (V, E)$ が与えられた際，その頂点数を n，辺数を m で表すことにする．また，頂点 $i, j \in V$ 間を結ぶ辺を $ij \in E$ と表す．グラフ G と平面上の点集合 $\mathbf{p} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\} \subset \mathbb{R}^2$ が与えられた際，頂点 $i \in V$ を点 $\mathbf{p}(i) = \mathbf{p}_i$ へ，辺 ij を伸び縮みのできない線分 $\mathbf{p}_i\mathbf{p}_j$ へと埋め込んで得られる幾何グラフを棒とジョイントからなる**フレームワーク** (bar-and-joint framework) と呼び，$G(\mathbf{p})$ と表される．

フレームワーク $G(\mathbf{p})$ に対して，点集合 $\mathbf{p}$ の平面上での移動を考えるために，各点へ速度ベクトル $\mathbf{u} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n\}$ を割り当てる．このとき，辺長が一定である条件から等式系

$$(\mathbf{p}_i - \mathbf{p}_j) \cdot (\mathbf{u}_i - \mathbf{u}_j) = 0, \quad {}^{\forall} ij \in E \tag{1}$$

が得られる．この等式系を満たす $\mathbf{u}$ を**微小変形** (infinitesimally motion) と呼ぶ．またベクトル $\mathbf{u}_x = [\ 1\ \ 0\ \ \ldots\ \ 1\ \ 0\ ]$, $\mathbf{u}_y = [\ 0\ \ 1\ \ \ldots\ \ 0\ \ 1\ ]$, $\mathbf{u}_r = [\ -y_1\ \ x_1\ \ \ldots\ \ -y_n\ \ x_n\ ]$ の線形結合で得られる微小変形を**自明な微小変形** (trivial infinitesimal motions) と呼び, 等式系の解が自明な微小変形のみの時，フレームワークは (微小変形に対して) **剛である** (infinitesimally rigid) という．またそれ以外の時，フレームワークは (微小変形に対して) **柔である** (infinitesimally flexible)



(a)

(b)

図 1　(a) 剛なフレームワークと (b) 柔なフレームワーク．

という．図 1(a) は剛なフレームワークの，図 1(b) は柔なフレームワークの例である．(b) のフレームワークについて太線で示しされた矢印が自明でない微小変形を表している．

式 (1) の等式系を行列表現した際得られる $m \times 2n$ の行列 $R_G(\mathbf{p})$ を**剛性行列** (rigidity matrix) と呼ぶ．上で示した自明な微小変形 $\mathbf{u}_x, \mathbf{u}_y, \mathbf{u}_r$ に対して，それぞれが $R_G(\mathbf{p})\mathbf{u}^\top = 0$ の解であることは容易に確かめることができる．このことから任意のフレームワークに対して $\dim \ker R_G(\mathbf{p}) \geq 3$ であり，フレームワークが剛である事（つまり，自明な変形しか持たない事）と $\mathrm{rank}\ R_G(\mathbf{p}) = 2n - 3$ が成り立つことは等価である．

ある一般の点位置 $\mathbf{p}$ 上のフレームワーク $G(\mathbf{p})$ が剛であるとき，グラフ G は剛 (rigid) であるという．一般の点配置 $\mathbf{p}$ 上での剛性行列 $R_G(\mathbf{p})$ に対して，$R_G(\mathbf{p})$ の各行ベクトルが線形独立のとき，各行ベクトルに対応する G の各辺が独立である (edge independence) という．これにより辺集合から構成される独立集合族を定義することができ，辺集合を台集合とした **(一般) 剛性マトロイド** (generic rigidity matroid) が得られる．

既に述べた通り $\mathrm{rank}\ R_G(\mathbf{p}) \leq 2n-3$ であることから，極大独立集合つまりマトロイドの基の要素数は $2n - 3$ である．また G が基である時，G は最小の辺数で剛 (minimally rigid) であり，**静定グラフ** (isostatic graph) 又は**ラーマングラフ** (Laman graph) と呼ばれる．最小の辺数で剛であるグラフについて、以下の定理が成り立つ事が Laman [10] によって示されている．

定理 3 (Laman, [10]) 頂点数 $n \geq 2$ のグラフ G が最小の辺数で剛であるための必要十分条件は以下の条件 (1)(2), (Laman counts) が成り立つことである．

(1)　$m = 2n - 3$,

(2)　すべての 2 頂点以上の頂点部分集合の誘導部分グラフにおいて，$m' \leq 2n' - 3$.

ラーマングラフ G が一般の点配置 $\mathbf{p}$ 上に実現された時，$G(\mathbf{p})$ は最小の部材数で剛なフレームワーク (minimally rigid framework) であり，**静定フレームワーク**又は**ラーマンフレームワーク**という．

3　辺制約付き三角形分割

ここから先においては，平面頂点集合 P 上の幾何グラフのみを扱う．そのため平面上の点 $p_i \in P$ を単に頂点 i，線分 $p_i p_j$ を辺 ij と記すことにする．平面上の n 個の頂点からなる頂点集合 P 上に三角形の個数 (面数) k の三角形分割 T が与えられている．T の角度ベクトル (angle vector) とは，T の三角形の $3k$ 個の内角を非減少順に並べたベクトルである．F を P 上の無交差な線分の集合とする．この時，F を部分集合として含むような三角形分割を ***F*-制約付き三角形分割** (F-constrained triangulation) という．平面上の 2 点 a, b に関して，線分 ab が F の要素と端点以外で交差しない時，a, b は**可視** (visible) である．また，線分 ab と点 c に対して，三角形 abc が F の要素と F の端点以外で交差しない時，ab と c は可視である．

定義 2 (Definition 1, [6]) ***F*-制約付き Delaunay 三角形分割** (F-constrained Delaunay triangulation or CDT) が点 $a, b \in P$ 間を結ぶ線分 ab を含む必要十分条件は，点 a, b が可視であり，ab と可視な点 $c \in P$ を含まないような a, b 上を通る円が存在することである．

点集合 P の凸包上に存在しない (内部の) 辺 $ac \notin F$ は 2 つの三角形 acb 及び acd と接している．四角形 $abcd$ が凸四角形であるとき，辺 ac を取り除き，辺 bd を加えることで新たな三角形分割が得られる．特に三角形 abc の外接円が点 c を含むとき，辺 ac を辺 bd へと変換する操作を **Delaunay フリップ** (Delaunay flip) 又は **D-フリップ**と呼ぶ. この時，ac を ***F*-illegal**，bd を ***F*-legal** と呼ぶ．D-フリップの前後で三角形の角度ベクトルが増加していることが容易に示せる．このことか

ら以下の事実が成り立つ．

定理 4 (Theorem 1, [6]) CDT は P 上の F-制約付き三角形分割の中で辞書式順序で最大の角度ベクトルをもつ．

定理 5 (Lemma 4, [6]) 任意の F-制約付き三角形分割は任意の順序の $O(n^2)$ 回の D-フリップで CDT へ移ることができる．

4 制約付き無交差ラーマンフレームワーク

P を平面上の頂点集合とする．無交差な線分集合 F が P の完全グラフ K_n の辺を台集合とする剛性マトロイドの独立集合の時，F を含む P 上のラーマンフレームワークを **F-制約付きラーマンフレームワーク**と呼ぶ．T を P 上の F 制約付き Delaunay 三角形分割とする．また，T の辺集合も便宜上 T で表す．このとき次の補題が剛性マトロイドの T への**制限** (restriction) から成り立つ (例えば [13])．

補題 1 P 上の無交差な辺集合 F が K_n を台集合とする剛性マトロイドの独立集合である時，全ての F 制約付き三角形分割 T は，少なくとも一つの F 制約付きラーマンフレームワークを部分集合として含み，T の辺集合はマトロイドを構成する．

証明 三角形分割は静的に剛であり，その辺集合は剛性マトロイドの基 B を含んでいる (例えば [14])．よってマトロイドの T への制限より T の辺集合はマトロイド $\mathcal{M}$ を構成する．F は $\mathcal{M}$ の独立集合であるので，$B - F$ の辺を加えることにより T の部分集合の F 制約付きラーマンフレームワークへ拡張することができる．■

定義 3 CDT の部分集合である F-制約付きラーマンフレームワークを **F-制約付き Delaunay ラーマンフレームワーク** (CDLF) と呼ぶ．

$\mathcal{L}$ を P 上の F 制約付き無交差ラーマンフレームワークの集合とし，$\mathcal{DL} \subseteq \mathcal{L}$ を CDLF の集合とする．これから集合 $\mathcal{L}$ が L-フリップを用いて連結であることを示そう．まず以下に示すように $L \in \mathcal{L} - \mathcal{DL}$ が CDLF へ L-フリップを用いて変換可能であることを示す．

補題 2 $L \in \mathcal{L} - \mathcal{DL}$ とする．$O(n^2)$ 回の L-フリップで L を CDLF へ変換することができる．

証明は L を部分集合として含む補助三角形分割 $T(L)$ を用いておこなう．L-制約付き Delaunay 三角形分割 $T(L)$ とは，次の操作によって得られる三角形分割である．まず P の凸包の境界上の辺で L に存在しないものを L に加え，各面に対してその面内部の Delaunay 三角形分割を計算し，新たな辺を加える．この定義によって，次の事実が成り立つ．

事実 1 $L \in \mathcal{L}$ とする．この時，L-制約付き Delaunay 三角形分割 $T(L)$ における全ての F-illegal な辺は $L - F$ の辺である．

証明（補題 2） L は CDLF ではない場合，F-illegal な辺 ac を持ち，事実 1 から ac は $L - F$ の要素である．よって L フリップの削除する辺として ac を選び，L から取り除く．すると，補助三角形分割 $T(L - ac)$ は，$L - ac$ を必ず含み，補題 1 から $L' = L - ac + st \in \mathcal{L}$ となる辺 st が $T(L - ac)$ 内に必ず存在する．ac は F-illegal であるという事実から，$T(L)$ から $T(L - ac)$ へ補助三角形分割が更新される際，少なくとも 1 回 D-フリップが起こり補助三角形分割の角度ベクトルはこの L-フリップ後必ず増加する．よって定理 4 と 5 から，$O(n^2)$ 回のフリップ操作後，最大の角度ベクトルを有する F-制約付き Delaunay 三角形分割 $T(L'')$ が得られ，L'' は CDLF となる．■

辺 $e = ij, (i < j)$ と $e' = kl, (k < l)$ に対して，$i < k$ または $i = k$ で $j < l$ ならば $e \prec e'$，$i = k$ かつ $j = l$ ならば $e = e'$ と記し，辺の辞書式順序を定める．また辺集合 A に対して，A の中で辞書式順序の一番大きい要素及び一番小さい要素を $\max\{e \mid e \in A\}$ と $\min\{e \mid e \in A\}$ と記す．辞書式順序を示す辺リスト $E = \{e_1 \prec e_2 \prec \cdots \prec e_m\}$, $E' = \{e'_1 \prec e'_2 \prec \cdots \prec e'_m\}$ において，$e_i \neq e'_i$ となる最小の i で $e_i \prec e'_i$ が成り立つ時，E は辞書式順序で E' より小さいと定める．

証明（定理 1） 補題 2 より，$O(n^2)$ 回の L-フリップで $\mathcal{L} - \mathcal{DL}$ の要素から $L \in \mathcal{DL}$ へたどり着くことができる．L^* を CDLF のうち辞書式順序で最小の辺リストをもつラーマンフレームワークとする．このとき L から多くとも $n - 3$ 回の L-フリップを用いて L^* へ変換することができることを示そう．L の辺のうち辞書式順序で最大な辺 ac を取り除く．すると $L - ac$ は剛性マトロイドの極大要素とはなっていないので $L^* - L$ の要素 st を用いて $L' = L - ac + st$ へ拡張することができ

る．また L, L^* 共に CDT の部分集合であることから L' は必ず無交差なラーマンフレームワークとなる．オイラーの方程式から L は最大で $n-3$ 個の L^* に含まれていない辺をもつ．よって多くとも $n-3$ 回の L-フリップによって CDLF は L^* へと変換される． ■

5　アルゴリズム

定理 1 の証明に基づき，各ラーマンフレームワークに対して親子関係を定める関数 $f : \mathcal{L} - \{L^*\} \to \mathcal{L}$ を以下のように定める．

定義 4　$L \in \mathcal{L} - \{L^*\}$ に対して $L' = L - ac + st$ は L の親である．ここで ac 及び st は，

- $L \in \mathcal{DL}$ の時，$ac = \max\{e \mid e \in L - L^*\}$,
$st = \min\{e \in L^* - L \mid L - ac + e \in \mathcal{L}\}$,
- $L \in \mathcal{L} - \mathcal{DL}$ の時，
$ac = \max\{e \in T(L) \mid T(L)$ 上で F-illegal な辺 $\}$,
$st = \min\{e \in T(L - ac) - (L - ac) \mid L - ac + e \in \mathcal{L}\}$.

$L \in \mathcal{DL} - \{L^*\}$ であるか $L \in \mathcal{L} - \mathcal{DL}$ であるかに応じて，f を $f_1 : \mathcal{DL} - \{L^*\} \to \mathcal{DL}$ 及び $f_2 : \mathcal{L} - \mathcal{DL} \to \mathcal{L}$ と表記することにする．図 3 は $L \notin \mathcal{DL}$ に対する $f_2(L)$ の例である．図では，まず辞書順序で最大の F-illegal 辺 37 を取り除き，$T(L-37) - (L-37)$ 内における最小の辺 12 を加えることにより新たな無交差ラーマンフレームワークを得ている．

ある無交差ラーマンフレームワーク L' に隣接しているオブジェクトの局所探索を行う隣接関数, Adj, を以下のように定める．$e_1 \in L' - F, e_2 \in K_n - L'$ に対して，

$$Adj(L', e_1, e_2) := \begin{cases} L' - e_1 + e_2 & L' - e_1 + e_2 \in \mathcal{L} \text{ の時} \\ null & \text{それ以外の時.} \end{cases}$$

辺のペア (e_1, e_2) は $O(n^3)$ 個存在するので，L' は $O(n^3)$ 個のオブジェクトと隣接していることがわかる．$elist_{L'}$ 及び $elist_{K_n}$ を L' 及び K_n の辺を辞書式順序に並べた辺リストとし，$\delta(L')$ と $\delta(K_n)$ を $elist_{L'}$ と $elist_{K_n}$ の要素数とする．また $elist_{L'}(i)$ と $elist_{K_n}(i)$ はそれぞれの i 番目の要素を示し，$e_1 = elist_{L'}(i)$, $e_2 = elist_{K_n}(j)$ の時 $Adj(L', e_1, e_2)$ を $Adj(L', i, j)$ と記すことにする．このとき Avis and Fukuda [2, 3] の逆探索手法に基づき図 2 に制約付き無交差ラーマンフレームワークの列挙アルゴリズムを示す．

Algorithm F-制約付き無交差ラーマンフレームワークの列挙.

1: $L^* :=$ 辞書式順序で最小の辺リストからなる CDLF;
2: $L' := L^*$; $i, j := 0$; $Output(L')$;
3: **repeat**
4: 　**while** $i \leq \delta(L')$ **do**
5: 　　$i := i + 1$;
6: 　　**while** $j \leq \delta(K_n)$ **do**
7: 　　　$j := j + 1$;
8: 　　　**if** $elist_{L'}(i) \notin F$ かつ $Adj(L', i, j) \neq null$ **then**
9: 　　　　$L := Adj(L', i, j)$;
10: 　　　　**if** $f_1(L) = L'$ or $f_2(L) = L'$ **then**
11: 　　　　　$L' := L$; $i, j := 0$;
12: 　　　　　$Output(L')$;
13: 　　　　　**go to** 第 4 行;
14: 　　　　**end if**
15: 　　　**end if**
16: 　　**end while**
17: 　**end while**
18: 　**if** $L' \neq L^*$ **then**
19: 　　$L := L'$;
20: 　　**if** $L \in \mathcal{DL}$ **then** $L' := f_1(L)$;
21: 　　**else** $L' := f_2(L)$;
22: 　　$Adj(L', i, j) = L$ となる整数対 (i, j) を求める;
23: 　　$i := i - 1$;
24: 　**end if**
25: **until** $L' = L^*$, $i = \delta(L')$ かつ $j = \delta(K_n)$;

図 2　F-制約付き無交差ラーマンフレームワーク列挙アルゴリズム.

詳細は後述するが，[11] による剛性マトロイドの極大成分を保持するデータ構造を用いると f 及び Adj を $O(n^2)$ の計算時間で実行することができる．よって各 $O(n^3)$ 回の局所探索に対してアルゴリズムは Adj と f を実行することから計算時間は $O(n^5)$ となる．しかし局所探索を行う辺のペアを適切に特徴づけることにより定理 2 を示すことができる．証明には，以下の 2 つの補題を用いる．

補題 3　L' を CDLF，L を $L = Adj(L', e_1, e_2)$ を満たす CDLF とする．このとき $f_1(L) = L'$ が成立する必要十分条件は $e_1 \in L' - F$ と $e_2 \in K_n - L'$ が以下の条件を満たすことである．

(a)　$e_1 \in L^*$,
(b)　$e_2 \in T(L^*) - L^*$,
(c)　$e_1 \prec \min\{e \in L^* - L' \mid L' - e_1 + e \in \mathcal{L}\}$,
(d)　$e_2 \succ \max\{e \mid e \in L' - L^*\}$.

証明　(必要条件) $f_1(L) = L'$ が成り立つことから，f_1 を L に実行した際，e_1 及び e_2 は定義 4 の場合 1 での

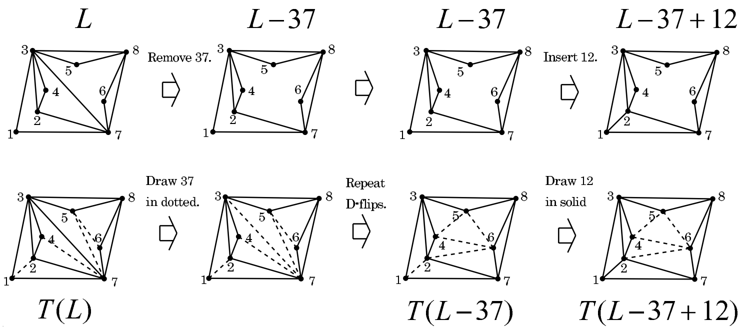


図 3　関数 f_2 の実行例．実線は L の辺を，点線は $T(L)$ のために新たに加えた辺を示している．

st (加えられる辺) 及び ac (除かれる辺) として選択され，$L' = L - ac + st$ が成り立つ．よって定義 4 から $e_2(= ac) \in L - L^*$ が成り立つ．また L は CDLF であり，$L \subseteq T(L^*)$ であることから，$e_2 \in T(L^*) - L^*$ が成り立つ．これは条件 (b) である．同様に，$e_1(= st) \in L^* - L \subset L^*$ から条件 (a) を得る．また $e_1 = st$ から，

$$L' - e_1 = (L - ac + st) - e_1 = L - ac \qquad (2)$$

を得る．$e' = \min\{e \in L^* - L' \mid L' - e_1 + e \in \mathcal{L}\}$ とする．このとき条件 (c) が成り立たないと仮定すると，$e' \prec e_1$ が成り立つ ($e_1 \in L' - F$ であることから $e' \neq e_1$ であることに注意)．よって式 (2) 及び $e' \prec e_1 = st \prec ac$ から，

$$\begin{aligned} e' &= \min\{e \in L^* - L' \mid L' - e_1 + e \in \mathcal{L}\} \\ &= \min\{e \in L^* - (L - ac + st) \mid L - ac + e \in \mathcal{L}\} \\ &= \min\{e \in L^* - L \mid L - ac + e \in \mathcal{L}\} \end{aligned}$$

となり，$f_1(L)$ を実行した際，f_1 は e' を新たに加える辺として選択し，$e_1 = st$ に矛盾する．最後に条件 (d) が成り立つことを示そう．$e'' = \max\{e \mid e \in L' - L^*\}$ とし，条件 (d) が成り立たない，つまり $e_2 \prec e''$ であると仮定する ($e_2 \notin L'$ から $e'' \neq e_2$ であることに注意)．このことから $st \prec ac = e_2 \prec e''$ であり，

$$\begin{aligned} e'' &= \max\{e \mid e \in L' - L^*\} \\ &= \max\{e \mid e \in (L - ac + st) - L^*\} \\ &= \max\{e \mid e \in L - L^*\} \end{aligned}$$

を得る．f_1 は e_2 ではなく e'' を取り除く辺として選択し，$e_2 = ac$ に矛盾する．

(十分条件)　条件 (b) から，$L = L' - e_1 + e_2 \in \mathcal{DL}$ が確かに成り立つ．条件 (a) の $e_1 \in L^*$ から条件 (d) は，

$$\begin{aligned} e_2 &\succ \max\{e \mid e \in L' - L^*\} \\ &= \max\{e \mid e \in (L + e_1 - e_2) - L^*\} \\ &= \max\{e \mid e \in (L - e_2) - L^*\} \end{aligned}$$

となり，$e_2 = \max\{e \mid L - L^*\}$ が成り立つ．従って，定義 4 から f_1 は e_2 を取り除く辺として選択する．またこのことから $L - ac = L' - e_1 + e_2 - ac = L' - e_1$ を得る．条件 (b) の $e_2 \notin L^*$ から条件 (c) は，

$$\begin{aligned} e_1 &\prec \min\{e \in L^* - L' \mid L' - e_1 + e \in \mathcal{L}\} \\ &= \min\{e \in L^* - (L + e_1 - e_2) \mid L - ac + e \in \mathcal{L}\} \\ &= \min\{e \in L^* - (L + e_1) \mid L - ac + e \in \mathcal{L}\} \end{aligned}$$

となる．よって $e_1 \in L^* - L$ であることから，上式から $e_1 = \min\{e \in L^* - L \mid L - ac + e \in \mathcal{L}\}$ を得る．従って，f_1 は e_1 を新たに加える辺として選択し，$f_1(L)$ は L' を返す．■

補題 4　$L' \in \mathcal{L}$ とし，L を $L = Adj(L', e_1, e_2)$ かつ $L \in \mathcal{L} - \mathcal{DL}$ を満たす無交差ラーマンフレームワークとする．このとき $f_2(L) = L'$ が成立する必要十分条件は $e_1 \in L' - F$ と $e_2 \in K_n - L'$ が以下の条件を満たすことである．

(a)　e_1 は三角形分割 $T(L')$ 上において F-legal,

(b)　$e_2 \in K_n - T(L')$,

(c)　$e_1 \prec \min\{e \in T(L') - L' \mid L' - e_1 + e \in \mathcal{L}\}$,

(d)　$e_2 \succ \max\{e \in T(L') \mid T(L')$ 上で F-illegal な辺 $\}$.

証明　(必要条件) $f_2(L) = L'$ が成り立つことから，f_2 を L に実行した際，e_1 及び e_2 は定義 4 の場合 2 での st (加えられる辺) 及び ac (除かれる辺) として選択され，$L' = L - ac + st$ が成り立つ．よって $e_1 = st$ であるから，

$$L' - e_1 = (L - ac + st) - e_1 = L - ac \qquad (3)$$

が成り立つ．定義 4 より，$st \in T(L - ac) - (L - ac)$ であり，事実 1 から辺 st は $T(L - ac)$ 上で F-legal である．よって st を $L - ac$ に新たに加えることによって得られる三角形分割 $T(L - ac + st) = T(L')$ においても st

は F-legal である．よって $e_1(=st)$ は $T(L')$ 上において F-legal であり，条件 (a) が得られる．また e_1 が $T(L')$ 上で F-legal であることから，

$$T(L') = T(L' - e_1) = T(L - ac) \tag{4}$$

であることがわかる．

$e' = \min\{e \in T(L') - L' \mid L' - e_1 + e \in \mathcal{L}\}$ とする．条件 (c) が成り立たないと仮定すると，$e' < e_1$ が成り立たなければならない．しかし，式 (3) と (4) から，

$$\begin{aligned} e' &= \min\{e \in T(L') - L' \mid L' - e_1 + e \in \mathcal{L}\} \\ &= \min\{e \in T(L - ac) - (L - ac + e_1) \mid L - ac + e \in \mathcal{L}\} \\ &= \min\{e \in T(L - ac) - (L - ac) \mid L - ac + e \in \mathcal{L}\} \end{aligned}$$

となることから，f_2 は e' を新たに加える辺として選択し，これは $e_1 = st$ に矛盾する．

次に辺 e_2 に関する条件を考える．定義 4 より，$e_2(=ac)$ は $T(L)$ 上において F-illegal な辺である．よって L から辺 $e_2 = ac$ を取り除いて得られる三角形分割 $T(L - ac)$ 上に e_2 は存在しない．さらに式 (4) から条件 (b) の $e_2 \notin T(L')$ を得る．最後に条件 (d) が成り立つことを示すために，$e'' = \max\{e \in T(L') \mid T(L')$ 上で F-illegal な辺 $\}$ とし，条件 (d) が成り立たないと仮定しよう．このことから $e_2 = ac < e''$ が成り立つ．式 (4) から，$e'' = \max\{e \in T(L - ac) \mid T(L - ac)$ 上で F-illegal な辺 $\}$ である．よって e'' は $T(L - ac)$ 上で F-illegal な辺なので，$e'' \in T(L)$ でありかつ e'' は $T(L)$ 上においても F-illegal な辺である．よって $e'' = \max\{e \in T(L) \mid T(L)$ 上で F-illegal な辺 $\}$ となり，f_2 は e'' を削除する辺として選択することになるので，これは $e_2 = ac$ に矛盾する．

(十分条件) 条件 (a) から，e_1 は $T(L')$ 上で F-legal な辺である．よって $T(L') = T(L' - e_1)$ を得る．もし e_2 が $T(L) = T(L' - e_1 + e_2)$ 上で F-legal な辺ならば，$T(L' - e_1 + e_2) = T(L' - e_1) = T(L')$ かつ $e_2 \in T(L')$ が得られ，これは条件 (b) に矛盾する．よって e_2 は $T(L)$ 上で F-illegal な辺であり，条件 (d) を用いて，$e_2 = \max\{e \in T(L) \mid T(L)$ 上で F-illegal な辺 $\}$ が得られる．こうして，f_2 は L から取り除く辺として e_2 を選択する．またこのことから $L' - e_1 = L - ac$ を得る．よって条件 (c) は，

$$\begin{aligned} e_1 &< \min\{e \in T(L') - L' \mid L' - e_1 + e \in \mathcal{L}\} \\ &= \min\{e \in T(L' - e_1) - L' \mid L' - e_1 + e \in \mathcal{L}\} \\ &= \min\{e \in T(L - ac) - (L - ac + e_1) \mid L - ac + e \in \mathcal{L}\} \end{aligned}$$

と変形され、条件 (a) の $e_1 \in T(L') = T(L' - e_1) = T(L - ac)$ から，$e_1 = \min\{e \in T(L{-}ac){-}(L{-}ac) \mid L{-}ac{+}e \in \mathcal{L}\}$ を得る．よって，f_2 は L に加える辺として e_1 を選択し，f_2 は L' を返す． ■

証明（定理 2） ある無交差ラーマンフレームワーク L' が与えられた際，図 2 の第 4 行から 17 行目までの操作が $O(n^3)$ の計算時間で実行可能である事を示す．具体的には，$O(n)$ 個の要素からなる $elist_{L'}$ の各要素 e_1 に対して，$O(n^2)$ の計算時間の前処理を行うことで以下が実行可能である事を示す．

- 各辺 $e_2 \in elist_{K_n}$ に対して，$Adj(L', e_1, e_2)$ がラーマンフレームワークを返すかを定数時間で判定．
- 各辺 e_2 に対して，$f_1(Adj(L', e_1, e_2)) = L'$ 又は $f_2(Adj(L', e_1, e_2)) = L'$ が成り立つか，つまり補題 3 又は補題 4 を e_1, e_2 が満たすかを定数時間で判定．

これらにより，各辺 $e_1 \in elist_{L'}$ に対して，第 6 行目から 16 行目までの while ループが $O(n^2)$ の計算時間で実行可能となり，全体の計算時間は $O(n^3)$ となる．

まず，各 $e_1 \in elist_{L'}$ に対して，e_1 が補題 3 及び補題 4 の各条件 (a) を満たすかを調べ，各辺にフラグを付ける．これにより e_1 が条件 (a) を満たすかを定数時間で判定できる．同様に，各 $e_2 \in elist_{K_n}$ に対しても補題 3 及び補題 4 の各条件 (b) を満たすかを示すフラグを付ける．この前処理は $O(n^2)$ 時間で実行でき，これによって e_2 が条件 (b) を満たすかを定数時間で判定することができる．また，L' 上の辞書式順序最大の F-illegal な辺と $L' - L^*$ 上の辞書式順序最大の辺は $O(n)$ 時間で計算することができ，これにより各 e_2 が条件 (d) を満たすかを定数時間で判定することができる．

次に，e_1 が補題 3 または補題 4 の条件 (c) を満たしているかを定数時間で判定するために，Lee, Streinu and Theran [11] によるデータ構造を用いる．辺 e_1 を取り除いた際得られる自由度 1 のフレームワーク上において，このデータ構造は $O(n^2)$ の前処理時間で，剛性マトロイドの極大成分を計算，保持し，ある 2 つの頂点間が同じ極大成分に属しているかを定数時間で答える *pair-find* をサポートする．このデータ構造を用いることで、線形時間で $e' = \min\{e \in L^* - L' \mid L' - e_1 + e \in \mathcal{L}\}$ 及び $e'' = \min\{e \in T(L') - L' \mid L' - e_1 + e \in \mathcal{L}\}$ を計算でき，これによって e_1 が条件 (c) を満たすかを定数時間で判定することができる．また，このデータ構造を

用いると，各 e_2 に対して，$Adj(L', e_1, e_2)$ がラーマンフレームワークを返すかどうかを定数時間で判定できる．

辺 $e_2 \in elist_{K_n}$ が与えられた際，各補題の条件 (b) から，$e_2 \in T(L^*) - L^*$ の時は $f_1(Adj(L', e_1, e_2)) = L'$ が成り立つかのみを，$e_2 \in K_n - T(L')$ の時は $f_2(Adj(L', e_1, e_2)) = L'$ が成り立つかのみを調べればよい．上述したとおり，どちらの操作も対応する各補題の必要十分条件を満たすかを調べればよく，それらは定数時間で可能である．

また剛性マトロイドの極大成分を保持するデータ構造を用いると，ある 2 つの頂点間が同じ極大成分に属しているかを定数時間で答える事ができることから，Adj 及び f はそれぞれ $O(n^2)$ の計算時間で実行可能である． ■

6 まとめ

本論では，$O(n^3)$ 計算時間，$O(n^2)$ 容量の制約付き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．本論で新たに示した手法は，[3] で述べられている平面上の無交差木の列挙にも拡張可能である．平面上の木の集合はグラフマトロイドの基を構成するので，剛性マトロイドの場合と同様に，F-制約付き三角形分割を用いた F-制約付き無交差木の列挙アルゴリズムの構築が可能である．

参考文献

[1] O. Aichholzer, G. Rote, B. Speckmann, and I. Streinu. The zig-zag path of a pseudo-triangulation. In *Proc. 8th Int. Workshop on Algorithms and Data Structures (WADS)*, LNCS 2748, pages 377–388, Ottawa, 2003. Springer Verlag.

[2] D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete and Computational Geometry*, 8:295–313, 1992.

[3] D. Avis and K. Fukuda. Reverse search for enumeration. *Discrete Applied Mathematics*, 65(1-3):21–46, March 1996.

[4] D. Avis, N. Katoh, M. Ohsaki, I. Streinu, and S. Tanigawa. Enumerating non-crossing minimally rigid frameworks. In *Proc. 12th Annual International Computing and Combinatorics Conference (COCOON 2006)*, Taipei, 2006

[5] S. Bereg. Enumerating pseudo-triangulations in the plane. *Comput. Geom. Theory Appl.*, 30(3):207–222, 2005.

[6] M. Bern and D. Eppstein. Mesh generation and optimal triangulation. *Computing in Euclidean Geometry, 2nd Edition, Du and Hwang eds.*, 23–90, 1992.

[7] S. Bespamyatnikh. An efficient algorithm for enumeration of triangulations. *Comput. Geom. Theory Appl.*, 23(3):271–279, 2002.

[8] H. Brönnimann, L. Kettner, M. Pocchiola, and J. Snoeyink. Enumerating and counting pseudo-triangulations with the greedy flip algorithm. In *Proc. of ALENEX*, Vancouver, 2005.

[9] A. Dumitrescu, B. Gärtner, S. Pedroni, and E. Welzl. Enumerating triangulation paths. *Comput. Geom. Theory Appl.*, 20(1-2):3–12, 2001.

[10] G. Laman. On graphs and rigidity of plane skeletal structures. *Journal of Engineering Mathematics*, 4:331–340, 1970.

[11] A. Lee, I. Streinu, and L. Theran. Finding and maintaining rigid components. In *Proc. Canad. Conf. Comput. Geom.*, Windsor, Canada, 2005.

[12] G. Rote, F. Santos, and I. Streinu. Expansive motions and the polytope of pointed pseudo-triangulations. In J. P. Boris Aronov, Saugata Basu and M. Sharir (eds), *Discrete and Computational Geometry - The Goodman-Pollack Festschrift*, Algorithms and Combinatorics, (Springer Verlag, Berlin, 2003,) 699–736.

[13] D. J. A. Welsh. Matroids: Fundamental Concepts, In *R.L.Graham, M.Grötschel, and L.Lovász eds. Handbook of Combinatorics Vo.I.* (North-Holland, 1995), 481-526.

[14] W. Whiteley. Matroids from discrete geometry In *Matroid Theory, J. Bonin, J. Oxley and B. Servatius eds.* AMS Contemporary Mathematics, 171-313, 1997