Laman グラフの高速数え上げ

玉谷 賢一[1,a) 山内 由紀子[2] 来嶋 秀治[2] 山下 雅史[2]

概要：グラフの辺を伸び縮みをしない棒，頂点をジョイントとするフレームワークとみなし，このフレームワークに（小さな）力を加えても変形しないとき，このグラフは剛であるという．二次元で剛なグラフのうち辺の数が極小であるものはラーマングラフと呼ばれる．本稿は pebble game および Henneberg 構成を用いたラーマングラフの高速な数え上げアルゴリズムを提案する．また，計算機上に実装し単純なアルゴリズムとの比較を行う．

キーワード：ラーマングラフ，Henneberg 構成，pebble game，グラフ同型性

Fast Algorithms for Enumeration of Laman Graphs

Abstract: Suppose a graph denotes a framework consisting if rods and joints by edge and nodes, respectively. If the framework does not transform its form by a little force, the graph is called rigid. A minimally rigid graphs in the two dimensional space is called Laman graph. This paper proposes fast algorithms for enumeration of Laman graphs based on the pebble game and the Henneberg construction. Then, we compare these algorithms with an exhaustive search by computer experiments.

Keywords: Laman graph, Henneberg construction, pebble game, graph isomorphism

1. はじめに

グラフの辺を伸び縮みをしない棒，頂点をジョイントとするフレームワークとみなし，このフレームワークに（小さな）力を加えても変形しないとき，このグラフは剛であるという．このようなフレームワークの剛性を組み合わせ的に特徴付ける理論を組み合わせ剛性理論という [2]．三次元以上でのフレームワークに対する一般的な特徴付けはまだ知られていないが，一方で二次元内での特徴付けが知られている．Laman[2] は平面上に置かれたフレームワークの剛性を扱い，辺の数が極小となるグラフの特徴付けを与えた．具体的には，グラフ $G = (V, E)$ が二次元上の極小な剛グラフであることの必要十分条件が $|E| = 2|V| - 3$ かつ任意の頂点集合 $V' \subset V$ について V' による誘導部分グラフの辺数が $2|V'| - 3$ であることを示した．ラーマンの条件を満たすグラフを**ラーマングラフ**という．

1 九州大学工学部電気情報工学科
Department of Electrical Engineering and Computer Science, Kyushu University
2 九州大学大学院システム情報科学研究院)
Graduate School of Information Science and Electrical Engineering, Kyushu University
a) tamatani@tcslab.csce.kyushu-u.ac.jp

本研究では，ラーマングラフの構成方法である Henneberg 構成と，ラーマングラフの高速な判定アルゴリズムである pebble game に注目したラーマングラフの高速数え上げアルゴリズムを提案する．まず，pebble game を用いた頂点ラベル付きラーマングラフの数え上げアルゴリズムを示し，計算機上に実装し比較を行う．次に，この結果をもとにするラベルなしのラーマングラフの数え上げアルゴリズムと Henneberg 構成を用いたラベルなしのラーマングラフの数え上げアルゴリズムを示し，計算機上に実装し比較を行う．

2. 準備

2.1 ラーマングラフ

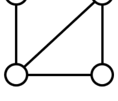

図 1

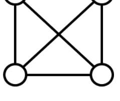

図 2

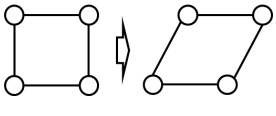

図 3

平面内で剛なグラフのうち辺が極小なグラフのことを

© 2015 Information Processing Society of Japan

ラーマングラフという [2]．図 1 はラーマングラフ．図 2 は剛なグラフであるが，辺を取り除いても剛なグラフであるため辺が極小でなくラーマングラフではない．図 3 は頂点の位置を動かすことで変形するため剛なグラフでなく，ラーマングラフではない．n 個の頂点集合は $2n$ 個の独立した自由度をを持っている．グラフが平面内で剛であるためには平行移動，回転移動を除いた自由度を制限するために $2n-3$ 個の辺を追加する必要がある．n' 頂点の部分グラフの辺数が $2n'-3$ より多い場合，明らかに独立でない自由度を制限する辺が存在する．以下の定理が成り立つ．

定理 1 ([1])．任意の部分グラフ $G' = (V', E')$ が $|E'|\leq 2|V'|-3$ を満たすとき，かつその時に限り，単純無向グラフ $G = (V, E)$ の辺集合は平面内で独立である．

この定理を用いてラーマングラフの条件を以下のように書くことができる．

定理 2 ([1], [2])．単純無向グラフ $G = (V, E)$ がラーマングラフである必要十分条件は以下の二つである．

(1) $|E| = 2|V| - 3$

(2) $|E[U]|\leq 2|U| - 3$ $(\forall U\subset V, |U|\geq 2)$

2.2 Henneberg 構成

定義 1 ([2])．次の二つの操作によって頂点数 k のグラフから頂点数 $k+1$ のグラフを構成することを *Henneberg* 操作という．

0-extension

新たな頂点を追加し，異なる二頂点との間に辺を追加する．

1-extension

辺を一つ取り除き，その両端及びそれ以外の一頂点と新たな頂点との間に辺を追加する．

これら二つの操作を繰り返し行い K_2 からグラフ G を構成していく一連の操作のこと G のを Henneberg 構成という．次の定理が成り立つことが知られている．

定理 3 ([2])．グラフ G が *Henneberg* 構成を持つことは G がラーマングラフであることの必要十分条件である．

2.3 Pebble covering

定義 2 ([1])．グラフ $G = (V, E)$ の各頂点は *pebble* を二つ持つものとする．各頂点は持っている *pebble* を接続している辺に置くことができる．このとき，すべての辺に *pebble* が置かれているような *pebble* の配置のことを *pebble covering* という．

2.4 Pebble game

ラーマングラフの条件を確認するためにはすべての頂点誘導部分グラフを列挙する必要があり，これには指数時間かかる．pebble game というアルゴリズムによってグラフ $G = (V, E)$ がラーマングラフであることを $O(|V|^2)$ で確認することができる．pebble covering について以下の定理が知られている．

定理 4 ([1])．グラフ $G = (V, E)$ の辺集合が独立であるとき, 新たな辺 e を 4 重辺にして追加した時の *pebble covering* が存在するならば e は E から独立である．

pebble game は pebble covering を持つグラフに新たな辺を追加した後の pebble covering を構成するアルゴリズムであり，ルールは以下の通りである [1]．

前提条件　グラフ $G = (V, E)$ の pebble covering が構成されているものとする．

(1) 新たな辺を $e = (v_1, v_2)$ とする．
(2) v_1 または v_2 に Pebble covering に使われていない（free な）pebble があればそれを用いて e をカバーする．
(3) そうでない場合，G の辺の pebble による向き付けに従って v_1 から free pebble を深さ優先探索する．
(4) free pebble をもつ頂点が見つかったら，v_1 からその頂点への有向道の向きを逆にするように pebble を再配置することで v_1 に free pebble を移動でき，e をカバーされる．
(5) 4 で free pebble をもつ頂点が見つからない場合 v_2 についてステップ 3 を実行する．
(6) free pebble をもつ頂点が見つからない場合，pebble covering の構成失敗．

探索の際に，一度訪れた頂点は再び探索しないようにすると $O(|V|)$ で新たな pebble covering が構成できる．定理 4 から pebble game を用いた以下の手順によって単純無向グラフ $G = (V, E)$ がラーマングラフであるか判定できる．

(1) $|E| = 2|V| - 3$ でなければ G はラーマングラフではなく終了．
(2) 多重グラフ $G' = (V', E')$ を考える．ただし初期状態は $V' = V, E' = \phi$ とする．
(3) G の辺を一つずつ順に選び G' に加える．
(4) pebble game によって新たな辺を 4 重辺にした時の G' の pebble covering を構成する．構成できないならば G はラーマングラフではい．
(5) 4 重辺にしたうちの 3 辺を取り除き，pebble はもとの頂点に戻す．
(6) 次の辺について 3 から繰り返す．
(7) すべての辺が追加されたならば G はラーマングラフである．

ステップ 3 ∼ 6 の繰り返し回数は $|V|$ であり，各繰り返しでは pebble game を 4 回実行するため $O(|V|^2)$ である．

3. 頂点ラベル付きラーマングラフの数え上げ

3.1 素朴なアルゴリズム

定理 2 の条件を用いる．頂点数 n の完全グラフの辺数は $\binom{n}{2} = n(n-1)/2$ 通りある．条件 (1) から使われる辺は

© 2015 Information Processing Society of Japan

$2n-3$ 本であり，選び方は $\binom{n(n-1)/2}{2n-3}$ 通りである．この各選び方について頂点誘導部分グラフは 2^n-2 通りあり，これらのすべての誘導部分グラフについて条件 (2) を満たせば 1 数え上げる．

3.2 高速アルゴリズム (1)

ラーマングラフの判定を pebble game によって行い，さらに分岐の際にそれ以降の分岐の形が同じになるものはまとめて計算する．

3.2.1 高速化のアイデア (1-a)

頂点を番号 ($0\sim n-1$) によってラベル付けする．初期状態で各頂点間に辺はないものとして以下の手順で辺を追加していく．

(1) 番号が小さい頂点から順に辺を追加する．
(2) 番号が小さい頂点から番号が大きい頂点に辺を追加する．

ここで以下の定理が成り立つことを示す．

定理 5. 番号 k の頂点に辺を追加したとき，同型なグラフ A,B が得られたとする．このとき A の $0\sim k$ 番の頂点すべてが B の $0\sim k$ 番の頂点へ対応するような同型写像が存在するならば A,B から同数のラーマングラフが得られる．

Proof. この同型写像では，A の $k+1\sim n-1$ 番の頂点すべてが B の $k+1\sim n-1$ 番の頂点へ対応している．A に頂点 $k+1$ 以降に辺を追加する際対応する辺を B に追加していく．この操作によって A から得られるラーマングラフと同数のラーマングラフを B から得られる．逆に B から得られるラーマングラフと同数のラーマングラフを A から得られる．よって A と B からは同数のラーマングラフが得られる． □

番号 0 の頂点に追加できる辺の数は $n-1$ 本であり，$0<k\le n-1$ 本追加する場合の付け加え方は $\binom{n-1}{k}$ 通りである．このとき番号 $1\sim n-1$ は孤立頂点であり，どの k 頂点を選んできても出来るグラフは同型であり，すべて 0 番の頂点を対応させるような同型写像が存在する．よってこれらはまとめて数え上げができる．図 4 は $n=6,k=2$ の場合であるが，定理 5 よりこれら $\binom{5}{2}=10$ 通りのグラフからはそれぞれ同数のラーマングラフが得られ，一番左の場合だけを計算し 10 倍すれば良い (便宜上辺にはラベルの小さい方から大きい方へ向き付けをしている)．

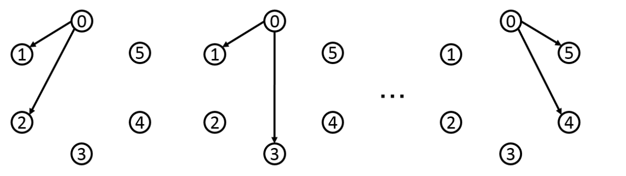

図 4

図 5 は番号 2 と 3 の頂点の間に辺を引く場合右側の $\binom{2}{1}=2$ 通りであるが，これらは同型であり，0 ～ 2 番の頂点同士を対応させる同型写像が存在するため，一つ目の場合だけを計算すればよい．

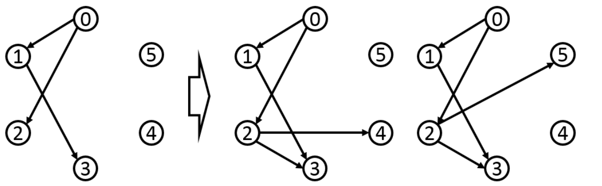

図 5

3.2.2 高速化のアイデア (1-b)

辺を追加する頂点が孤立頂点，次数 1 の頂点の場合新たに追加する辺は元の辺集合から独立であるため，この場合 pebble game を実行しなくても辺が追加できる．

3.3 高速アルゴリズム (2)

高速アルゴリズム (1) では同型性判定を用いずとも明らかに同型となるものをまとめて数え上げたが，ここでは同型性判定を用いる．頂点数 n のグラフの同型性判定は虱潰しで行おうとすると頂点の対応は $n!$ 通りあるため時間がかかる．そこで次数に注目して頂点を対応させる．同型性判定は頂点を次数によりソートして，バックトラックで同じ次数の頂点同士を対応させる．定理 5 から，番号 k の頂点に辺を追加したときに $0\sim k$ 番の頂点どうしを対応するグラフをそれぞれまとめて数え上げる．

3.4 計算機実験と考察

三つのアルゴリズムによりラーマングラフの数え上げを行い実行時間の比較を行った．

A: 素朴なアルゴリズム
B: 高速アルゴリズム (1)
C: 高速アルゴリズム (2)

表 1 実行結果

n	数え上げ	A	B	C
3	1	*	*	*
4	6	*	*	*
5	100	*	*	*
6	3355	0.031[sec]	0.015[sec]	0.015[sec]
7	190491	4.8[sec]	0.031[sec]	0.031[sec]
8	16445968	1300[sec]	0.53[sec]	0.59[sec]
9	2009020896		16[sec]	10[sec]
10	329848499025		600[sec]	230[sec]
11	70042422406825		26000[sec]	6800[sec]

計算時間の有効桁数は 2 とする．*は 0.01[sec] 未満. 実験環境は表 4 と同じである．

© 2015 Information Processing Society of Japan

高速アルゴリズムが素朴なアルゴリズムと比較して実行速度が改善されことが確認された．高速アルゴリズムでは同型なものをまとめて数え上げるため，実際に構成するラーマングラフの数はかなり少ない．以下の表に各アルゴリズム実際に構成するラーマングラフの数を示す．

表 2 実際に構成したラーマングラフの数

n	A	B	C
3	1	1	1
4	6	4	3
5	100	33	16
6	3355	434	137
7	190491	7993	1589
8	16445968	190998	24101
9		5634605	453128
10		198231686	10226820
11		8108675930	270003132

A は数え上げた個数と同じだけのラーマングラフを構成している．一方，B,C では，素朴なアルゴリズムに比べて頂点数が増すごとに構成するラーマングラフの数がかなり削減されていることがわかる．さらに，A はラーマングラフの判定を，ラーマンの条件に基づく指数時間かかる方を用いているが，B,C では pebble game を用いている．この二つのことからこれほどの差が生じたと考えられる．

A と B でも，構成するラーマングラフの数の差が大きく，その分が計算時間の差になっていると考えられる．また，頂点数 8 で C が B よりも遅くなっているのは，構成するラーマングラフの数の差に対して，同型性判定の計算時間が大きく出たためだと考えられる．

4. ラベルなしラーマングラフの列挙

4.1 Henneberg 構成に基づくアルゴリズム

Henneberg 操作の 0-extension および 1-extension を用いて，非同型な $n-1$ 頂点のラーマングラフの集合から非同型な n 頂点のラーマングラフを構成する．非同型な $n-1$ 頂点のラーマングラフすべてに対してありうるすべての Henneberg 構成を行い，その中から非同型なものを取り出せば良い．

4.2 高速アルゴリズム (2) を用いたアルゴリズム

3 節で示した高速アルゴリズム (2) ではラベル付きラーマングラフをある程度まとめて数え上げるため，ラベル付きの全列挙にはなっていないが，ラベルなしで考えると少なくとも非同型なラーマングラフ全通りは構成されているはずである．この中から非同型な物を取り出すことで非同型なラーマングラフの列挙をすることができる．

4.3 計算機実験と考察

以下の 3 通りの方法により非同型なラーマングラフの列挙を行った．確認のため 3 節で用いた改良した同型性判定に加えて虱潰しの同型性判定でも実行した．

A: Hennegerg 構成に基づくアルゴリズムと虱潰しによる同型性判定

B: Hennegerg 構成に基づくアルゴリズムと改良した同型性判定

C: 高速アルゴリズム (2) と改良した同型性判定

表 3 ラーマングラフの列挙

n	数え上げ	A	B	C
3	1	*	*	*
4	1	*	*	*
5	3	*	*	*
6	13	0.031[sec]	*	*
7	70	14[sec]	0.015 [sec]	0.63[sec]
8	608	7100[sec]	0.66[sec]	3.4[sec]
9	7222		95[sec]	770[sec]
10	110132		27000[sec]	

計算時間の有効桁数は 2 とする．*は 0.01[sec] 未満

表 4 実験環境

OS	Windows 8.1 Pro
CPU	Intel(R) Core(TM) i5-4570 CPU@3.20GHz
メモリ	DDR4 1600MHz 4.00GB × 1
使用言語	C 言語
時間の計測法	C 言語の clock 関数

これら 3 つの列挙方法で得られた非同型なラーマングラフの個数が一致したため $n=8$ のときの 608 は確からしい結果だと思われる．$n=9$ のときの 7222 でも 2 と 3 の二通りの方法で個数が一致していることが確認された．しかし，Newman[3] は頂点数 8 のとき 609 としており，異なる結果となった．いずれが正しい値であるか今後さらなる検証が必要である．

5. おわりに

頂点ラベル付きラーマングラフの数え上げ及びラベルなしラーマングラフの列挙の高速化を示した．この計算機実験を通じ OEIS に記されているラベルなしのラーマングラフの個数の列とは異なる結果を得た．いずれが正しい値であるかのさらなる検証が今後の課題である．

参考文献

[1] D. J. Jacobs and B. Hendrickson, An algorithm for two-dimensional rigidity percolation: the pebble game, Jounal of Computational Physics, 137(1997), 346–365.

[2] 加藤直樹 三角形分割とラーマングラフ:建築への応用, RAMP シンポジウム 2006, 135–152.

[3] D. S. Newman, Number of minimally rigid graphs on n vertices, The On-Line Encyclopedia of Integer Sequences, A227117, 2013. http://oeis.org/A227117

© 2015 Information Processing Society of Japan