Pebble Covering の数え上げ高速化

玉谷賢一 † 山内由紀子 ‡ 来嶋秀治 ‡ 山下雅史 ‡
† 九州大学工学部電気情報工学科 ‡ 九州大学院システム情報科学研究院

1 はじめに

平面内で剛なグラフのうち辺数が極小のグラフのことを Laman グラフという．グラフが Laman であることの必要充分条件は辺数が $2n-3$ かつ辺集合が独立であることである [1, 2]．Pebble Game というアルゴリズムによってグラフの辺集合が独立であるかどうかを判定することが出来る [1].

Pebble Game では，あるグラフについて Pebble Covering が存在するかを判定する．独立な辺集合に新たな辺を追加する際，その辺を 4 重辺にしたときの Pebble Covering が存在するならばその辺はもとの辺集合で独立である．これをグラフの全ての頂点で調べることでグラフの辺集合が独立であるかどうかを判定出来る [1].

本稿では Pebble Covering の数え上げの高速化を示す．

2 Pebble Covering の数え上げ

2.1 Pebble Covering

$G=(V,E)$ は無向グラフとする．初期状態では各頂点が 2 つの pebble をもっているものとして，以下の条件を満たす pebble の割当を Pebble Covering という [1].

- 各頂点は現在持っている pebble を接続している辺に置くことが出来る．
- 全ての辺の上に pebble がある．

これは G の辺 E に各頂点の出次数が高々 2 となる向き付けに等しい．一般にグラフ G に対して複数の Pebble Covering が存在する．また，$|E| < 2|V|$ の場合は pebble が全ては使われない（Pebble Covering の状態で頂点に pebble が残る）場合がある [1].

本稿では，以下頂点数 n，辺数 $2n$ の Pebble Covering の全通りの数え上げについて議論する．すなわち辺の pebble を置いている頂点から反対側への向き付けを考えると，この数え上げは出次数 2 の単純有向グラフの数え上げである．頂点にはラベルがあり，区別されるものとする．

2.2 数え上げアルゴリズム

2.2.1 素朴なアルゴリズム（虱潰し）

Algorithm 1 素朴なアルゴリズム

1: 頂点数 n の完全グラフの辺数は $\binom{n}{2}=n(n-1)/2$ 通り．このうち使われる辺は $2n$ 本であるから辺の選び方は $\binom{n(n-1)/2}{2n}$ 通り．
2: この各選び方について辺の向き付けは 2^{2n} 通り．
3: 各向き付けについて出次数が 2 であるならば 1 数え上げる．

2.2.2 分枝限定法

素朴なアルゴリズムは 2^{2n} 通りの向き付け全てを試すことや，Pebble Covering が存在しない辺の選び方についても同様に全通りの向き付けを試すため効率がよくない．

素朴なアルゴリズムでは辺数が $2n$ であるという条件に注目していたが，改良アルゴリズムでは各頂点の出次数が 2 であるという条件に注目して分枝限定法を用いる．

Algorithm 2 改良アルゴリズム

1: (初期条件) 全ての頂点は孤立点とする．
2: 任意の順に頂点を選ぶ．
3: 選んだ頂点から有向辺を引ける頂点（この頂点とまだつながっていない頂点）の個数を a とする．
4: $a \leq 1$ のとき，この頂点の次数を 2 にすることができないため失敗．
5: $a \geq 2$ のとき，外向辺の引き方は $\binom{a}{2}$ 通りあり，このそれぞれに分岐し，次の頂点について 2 から繰り返す．
6: 最後の頂点の外向辺を引いたら 1 数え上げる．

2.3 分枝限定法の高速化

後の表に示す通り，この分枝限定アルゴリズムは素朴なアルゴリズムに比べて若干の改善にはなっているが，$n \geq 10$ ではどちらの手法でも現実的な時間では解けなかった．そこで，改良アルゴリズムの高速化について考える．アイデアは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数え上げするというものである．

2.3.1 高速化のアイデア (1)

改良アルゴリズムのステップ 3 で孤立点が二つ以上ある場合，次のような選び方はまとめて数え上げできる．

1. 孤立点を 1 つ，孤立点でない頂点から 1 つ選ぶ場合.
2. 孤立点を 2 つ選ぶ場合.

1 のとき孤立点以外の頂点の方を固定すると，どの孤立点を選んでもグラフが同型になる．2 も同様にどの 2 頂点を選んできてもグラフが同型になる．

同型のグラフはそれ以降の分岐の形が同じになるような順で頂点を選ぶことができるため，まとめて数え上げができる．

2.3.2 高速化のアイデア (2)

改良アルゴリズムのステップ 3 で出次数 2 の頂点が 2 つ以上ある場合，次のような選び方はまとめて数え上げできる．

1. 出次数 2 の頂点を 1 つ，出次数 0 の頂点から 1 つ選ぶ場合
2. 出次数 2 の頂点を 2 つ選ぶ場合

1 のとき出次数 0 の頂点の方を固定すると，どの出次数 2 の頂点を選んでもこれ以降この辺を参照する操作は出てこない（改良アルゴリズムでは出次数 2 の頂点の間の辺を参照するような操作はない）ため，これ以降の分岐の形が同じになりまとめて数え上げができる．2 のときも同様である．

2.3.3 高速化アルゴリズム

Algorithm 3 高速化アルゴリズム

1: (初期条件) 全ての頂点は孤立点とする．
2: 任意の順に頂点を選ぶ．
3: 選んだ頂点から有向辺を引ける頂点（この頂点とまだつながっていない頂点）のうち孤立点を a 個、出次数 2 の頂点を b 個、それ以外の頂点を c 個とする．
4: $a+b+c \leq 1$ のとき，この頂点の次数を 2 にすることができないため失敗．
5: $a+b+c \geq 2$ のとき，外向辺の引き方は次のように場合分けできる．
  - a の頂点から 2 つ選ぶ場合:$\binom{a}{2}$ 通りの選び方のうち 1 通りを計算し，出てきた Pebble Covering の個数を $\binom{a}{2}$ 倍する．
  - b の頂点から 2 つ選ぶ場合:$\binom{b}{2}$ 通りの選び方のうち 1 通りを計算し，出てきた Pebble Covering の個数を $\binom{b}{2}$ 倍する．
  - c の頂点から 2 つ選ぶ場合:まとめての数え上げはできないので $\binom{c}{2}$ 通り全てに分岐する．
  - a の頂点から 1 つ，b の頂点から 1 つ選ぶ場合:ab 通りの選び方のうち 1 通りを計算し，出てきた Pebble Covering の個数を ab 倍する．
  - a の頂点から 1 つ，c の頂点から 1 つ選ぶ場合:c の頂点それぞれについて分岐し，a の頂点は a 通りの選び方のうち 1 通りを計算し，出てきた Pebble Covering の個数を a 倍する．
  - b の頂点から 1 つ，c の頂点から 1 つ選ぶ場合:c の頂点それぞれについて分岐し，b の頂点は b 通りの選び方のうち 1 通りを計算し，出てきた Pebble Covering の個数を b 倍する．

6: 最後の頂点の外向辺を引いたら 1 数え上げる．

3 計算機実験

実行環境　CPU:2.4GHz Intel Core i5/RAM:8 GB(DDR3)/OS:OS X 10.9.4
実行時間は三回の平均値（ただし*は 1 度の実行の値）．計測は $clock()$ 関数を使用．有効桁数 2．

n	Pebble Covering	1	2	3
5	24	0.000006[sec]	0.000026[sec]	0.000013[sec]
6	14490	0.0034[sec]	0.0015[sec]	0.000064[sec]
7	4590360	1.7[sec]	0.21[sec]	0.00042[sec]
8	1436893710	800[sec]	45[sec]	0.0028[sec]
9	502451412120		13000[sec]*	0.021[sec]
10	203034577143636			0.16[sec]
11	95515499986847640			1.4[sec]
12	52206557635168560360			12[sec]
13	32969338228517009014080			110[sec]
14	23894519022720589913502690			1100[sec]
15	19736460638800406069301676944			12000[sec]*

4 まとめ

実際に高速化によって素朴なアルゴリズムでは現実的な時間では計算できなかったところまで数え上げをすることができた．今後の課題は，同様の改良を用いて Laman グラフの数え上げの高速化を実現することである．

5 参考文献

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.