

マトロイド被覆問題に対する発見的的手法

青木一正 大館陽太 山崎浩一

群馬大学 工学部 情報工学科
〒376-8515 群馬県 桐生市 天神町 1-5-1

概要 マトロイド被覆問題とは、グラフの樹相度を求める問題を一般化した問題である。マトロイド被覆問題は多項式時間で解けるものの、その時間計算量は決して低くない。近年、川野等は既に知られていたグラフの樹相度 (edge arboricity) に対する線形時間で動作する 2 倍近似アルゴリズムを、マトロイド被覆問題に対する発見的アルゴリズムに拡張した。本研究ではこの川野等が提案したアルゴリズムを、cographic マトロイドと strict gammoid と呼ばれる 2 つのマトロイド上の被覆問題に対して実装し、実験的評価を行った。

Heuristic algorithms for the matroid covering problem

Kazumasa Aoki Yota Otachi Koichi Yamazaki

Department of Computer Science Gunma University
1-5-1 Tenjin-cho, Kiryu zip:376-8515, Gunma, Japan

abstract The matroid covering problem can be considered as a generalization of edge arboricity problem. It is known that the matroid covering problem can be solved in polynomial time, but the time complexity is still high. Kawano et al. have recently proposed a heuristic algorithm for the matroid covering problem by generalizing a linear time approximation algorithm with performance ratio 2 for edge arboricity problem. In this paper, we implement and evaluate the heuristic algorithm for cographic matroids and strict gammoids.

1 はじめに

$G = (V, E)$ をあるグラフとする。グラフ G を (または辺集合 E を) 森に分割するとは、 $G_i = (V, E_i)$ が森となるように辺集合 E を分割すること、すなわち、 $E = \cup_{1 \leq i \leq k} E_i$ 、 $E_i \cap E_j = \emptyset$ となる E の分割 E_1, \dots, E_k を求めることである。分割の個数 k は一般に複数の値をとり得るが、それらの値のうちで最も小さい値を G の樹相度と呼ぶ。グラフの樹相度は、 $O(|V||E| \log |V|)$ 時間で最適解を、[5]、線形時間で最適解の 2 倍以下の近似解を求めることができる [3, 1]。

グラフの樹相度を求める問題は、マトロイド被覆問題と呼ばれる問題に一般化することができる。マトロイドとは線形独立/従属性から生じる離散的構造を抽象化したもので、台集合と呼ばれるある集合 S と、 S の部分集合で独立であるものを集めてできた独立集合の族 \mathcal{I} からなり、 $M = (S, \mathcal{I})$ と一般に表記される。マトロイド被覆問題とは、台集合 S を独立な集合をなるべく少ない分割の個数で分割する問題で、その最小の分割の個数を分割数と呼ぶ。マトロイドに対するアルゴリズムを考えると、一般にあるオラクルへの、特に独立性を判定するオラクルへの質問回数を物差しとして用いる。マトロイド被覆問題を解くには、独立性を判定するオラクル対し、高々 $O(|S|^3 \log |S|)$ 回質問すればよいことが知られている [9]。

グラフの樹相度に対して線形時間の近似アルゴリズムの存在を考えると、マトロイド被覆問題に対して線形時間もしくは比較的少ない時間計算量での近似アルゴリズムの存在を期待することは自然である。また、マトロイド被覆問題に対する時間計算量は、多項式時間ではあるものの、低いとは言い難く、厳密解を犠牲にして高速性を求めるアルゴリズムの開発は重要であると考えられる。近年、川野等は樹相度を求める問題に対する線形時間近似アルゴリズムを、マトロイド被覆問題に対する発見的アルゴリズムに拡張した [7]。本研究ではこの川野等が提案したアルゴリズムを、cographic マトロイドと strict gammoid と呼ばれる 2 つのマトロイド上の被覆問題に対して実装し、実験的評価を行う。

本論文では、マトロイドに関して一般的に用いられる表記 [8, 10] を使用する。定義や基本的な性質等については、紙面の都合上割愛する。

2 川野等の発見的アルゴリズム

本節では、マトロイド被覆問題に対する川野等が提案した発見的アルゴリズムについて説明する。川野等のアルゴリズムは FMA (Flat Matrix Algorithm) と呼ばれ、与えられたマトロイドから生成されるフラット行列と呼ばれる行列を入力としている。まずフラット行列について説明する。

2.1 フラット行列

行列 $A = \{a_{i,j}\}$ がマトロイド M のフラット行列であるとは、 A が以下の条件を満たすときをいう、すなわち、

- A の各列は S の各要素の対応している、
- A の各行は M のコサーキットまたはコサーキットの和集合に対応している、
- A のどの列ベクトルも零ベクトルではない、
- 列 j に対応する要素が、行 i に対応するコサーキット (またはそれらの和集合) に属するならば $a_{i,j}$ は値 1 を、そうでなければ 0 をとる。

与えられたマトロイドに対して一般にフラット行列は一意には決まらない。どのフラット行列を選ぶかにより実行可能解の精度にも違いが生じる。また後述するとおり接続リストを用いてフラット行列を表現した場合、川野等のアルゴリズム FMA はマトロイドの要素数の線形時間で動作する。従って要素数が非常に大きいマトロイドに対しては、FMA 自体の処理時間よりもむしろフラット行列の生成にかかる時間の方が問題となる。独立オラクルに $O(\rho(S)|S|^2)$ 回質問することでフラット行列は生成できる、ここで S はマトロイドの台集合である。マトロイドの

種類が決まれば，その種類に特化した方法で，より高速にフラット行列を生成することが，多くの場合可能となる．

2.2 川野等のアルゴリズム FMA

$F_1 \supset F_2 \supset \dots \supset F_s$ をフラットのある単調減少列とする．このとき，任意の $1 \leq i \leq s-1$ に対し $|P \cap (F_i - F_{i+1})| \leq 1$ を満たす $P \subseteq S$ は独立集合であることが保証される．川野等はこの性質を利用し，以下に示すアルゴリズム FMA を提案した [7]．

Algorithm FMA (Flat Matrix Algorithm)

input: a flat matrix A of a matroid $M = (S, \mathcal{I})$.

output: independent sets covering S

```

/* First Part */
1:  $A_0 := A; i := 0;$ 
2: while  $A_i$  has a row do
3: begin
4:   find a minimum weight row vector  $r_i$  of  $A_i$ ;
5:    $A_{i+1} := SM(A_i, r_i); D_i := \{e : e \in A_i - A_{i+1}\}; i := i + 1;$ 
6: end;
/* Second Part */
7:  $cnt := i; j := 0;$ 
8: while  $\exists h$  such that  $D_h \neq \emptyset$  do
9: begin
10:   $j := j + 1; P_j := \emptyset;$ 
11:  for  $i := 1$  to  $cnt$  do
12:    if  $D_i \neq \emptyset$  then
13:      begin
14:        choose any  $x \in D_i; D_i := D_i - \{x\}; P_j := P_j \cup \{x\};$ 
15:      end
16:    end;
17:  output  $P_k (1 \leq k \leq j);$ 

```

川野等のアルゴリズム FMA は 2 つの部分に分かれていて，最初の部分はフラット行列からフラットの単調減少列 $A_1 \supset A_2 \supset \dots \supset A_s$ を作成し，2 番目の部分では S を， $|P_j \cap (A_i - A_{i+1})| \leq 1$ となるように，幾つかの独立集合 P_j に分割する．

川野等の提案する方法で本質的に重要な点は，フラットの単調減少列 $F_1 \supset F_2 \supset \dots \supset F_s$ でより良いものを選ぶことにある．川野等の方法では $\max_i |F_i - F_{i+1}|$ が分割数になるため，この値が小さくなるようなフラットの単調減少列を求めることが重要になる．

川野等の方法では，前処理で求めたフラット行列からフラットの単調減少列 $A_1 \supset A_2 \supset \dots \supset A_s$ を計算した．これとは別に，前処理無しでその都度毎にフラットを求めてゆき，フラットの単調減少列 $F_1 \supset F_2 \supset \dots \supset F_s$ を計算する方法も考えられる．例えば上述した本質的に重要な点を考慮すると次のような方法が考えられる，すなわち，単調減少列を求めるときに今現在のフラットよりも真に小さいフラットで，かつなるべく大きいフラットを求めてゆく方法，つまり極大なフラットの鎖を求めてゆく方法である．フラットの補集合がコサーキットの和集合に

対応することからこの方法は，なるべく小さいコサーキットを求めてゆく方法と言い換えることができる．このその都度毎になるべく小さいコサーキットを求めてゆく方法を，FMA 無しの方法 (またはアルゴリズム) と呼ぶことにする．

3 実験結果

本節では，川野等の発見的アルゴリズム FMA を，cographic マトロイドと strict gammoid の2つのマトロイドに適用し，実験的な評価を行う．比較対象として，貪欲的に独立集合をランダムに選び，取り去るという操作を繰り返すアルゴリズムを考える．そのようなアルゴリズムをここでは，SGA (straight greedy algorithm) と呼ぶことにする．実験環境は，CPU: Pentium 4 (3.0GHz), RAM: 1GB, OS: Windows XP である．

3.1 Cographic マトロイド

3.1.1 FMA 有りの方法

Cographic マトロイドでのコサーキットは，グラフでの閉路に対応する．つまり，基本タイセット行列をフラット行列とみなすことができ，それは $O(|V|^2)$ で求めることができる．例えば図1のグラフ G に対して，太線で示した全域木を考えると，表1に示すような基本タイセット行列，すなわちフラット行列が得られる．

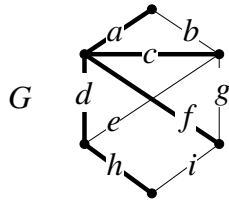


図 1: グラフ G .

	a	b	c	d	e	f	g	h	i
$C(b, B)$	1	1	1	0	0	0	0	0	0
$C(e, B)$	0	0	1	1	1	0	0	0	0
$C(g, B)$	0	0	1	0	0	1	1	0	0
$C(i, B)$	0	0	0	1	0	1	0	1	1

表 1: G に対する基本タイセット行列 (フラット行列)

このフラット行列により，例えば $\{a, d, f, h\}$, $\{b, e, g, i\}$, $\{c\}$ の3つの独立集合が得られる．

3.1.2 FMA 無しの方法

ここではFMA 無しの方法，すなわち，なるべく小さいコサーキットをその都度毎に求めてゆく方法について考える．FMA 無しの方法を cographic マトロイドに適用すると，cographic のコサーキットがグラフの閉路に対応することから，その時点で最小の閉路をその都度求めてゆく方法となる．グラフの最小閉路は $O(|V||E|)$ 時間で求めることができるが [6]，それを繰り返すため，FMA 有りよりは計算コストはかかる．具体的には以下のようなアルゴリズムとなる．

Algorithm COGRAPHIC-MINCYL

input: a graph $G = (V, E)$ without bridges

output: an upper bound of covering number of $M^*(G)$

```
1:  $m := 0$ ;  
2: while  $E(G) \neq \emptyset$  do  
3:   begin  
4:     find a cycle  $C$  with minimum length of  $G$ ;  
5:     if  $|C| > m$  then  $m := |C|$ ;  
6:      $G := G/C$ ; (i.e. the new  $G$  is obtained from the old  $G$  by contracting the edges of  $C$ )  
7:   end;  
8: output  $m$ ;
```

3つのアルゴリズム, FMA, COGRAPHIC-MINCYL, COGRAPHIC-SGA の比較を行う。テストデータの作成手順は次の通りである: ランダムグラフを生成し, ブリッジがあるか調べ, ブリッジがないグラフのみ使用する。COGRAPHIC-SGA は COGRAPHIC-MINCYL よりかなり速いので, 実行時間が同じくらいになるまで COGRAPHIC-SGA を繰り返し実行した (COGRAPHIC-SGA はランダムにスパニングツリーを採用するので決定的ではない)。実験には 500 頂点のグラフ 100 個と, 1000 頂点のグラフ 100 個を使用した。結果は表 2 の通りである。表 2 から分かる通り, COGRAPHIC-MINCYL は COGRAPHIC-SGA と同様によく働くが, FMA はよく働かない。

	MCL	SGA	FMA
平均	3.00	3.02	3.49
NB	100	98	51

	MCL	SGA	FMA
平均	3.00	3.00	3.68
NB	100	100	32

表 2: 500 頂点のグラフの結果 (左側) と 1000 頂点のグラフの結果 (右側) (NB は一番良い解が出た回数, MCL は COGRAPHIC-MINCYL, SGA は COGRAPHIC-SGA を表す)。

3.2 Strict gammoid

ここでは, FMA を strict gammoid に適用する。例を通して説明する。図 2 は 9 個の基地局からなるネットワークを表しており, 頂点は i ($1 \leq i \leq 9$) は基地局 i を, 有向辺 (i, j) は基地局 i から基地局 j へデータ転送可能であることを表している。基地局は, データを処理できるサービス局と処理できない非サービス局の 2 種類に分けられる。図 2 では, $1, 2, \dots, 6$ が非サービス局で, $7, 8, 9$ がサービス局である。各基地局 i にはそれぞれの仕事 t_i が割り当てられていて, 仕事 t_i を処理するには基地局 i からあるサービス局までデータを転送しなければならない。つまりグラフ D 上でいうなら, 基地局 i からあるサービス局までの有向パスが存在しなければならない。さらに, 各基地局は (自分自身への転送も含めて考えて) 一度に一つの基地局にしかデータを転送できない。従って, もし 2 つの仕事 t_i と t_j を同時に処理したい場合は, 基地局 i からあるサービス局への有向パス P_i と基地局 j からある別のサービス局への有向パス P_j が存在し, かつそれらが互いに点素でなければならない。例えば図 2 では $3, 4, 5$ は同時に処理が可能である。何故ならば, $3 \rightarrow 2 \rightarrow 9, 4 \rightarrow 7, 5 \rightarrow 8$ の 3 つの互いに点素な有向パスが存在するからである。

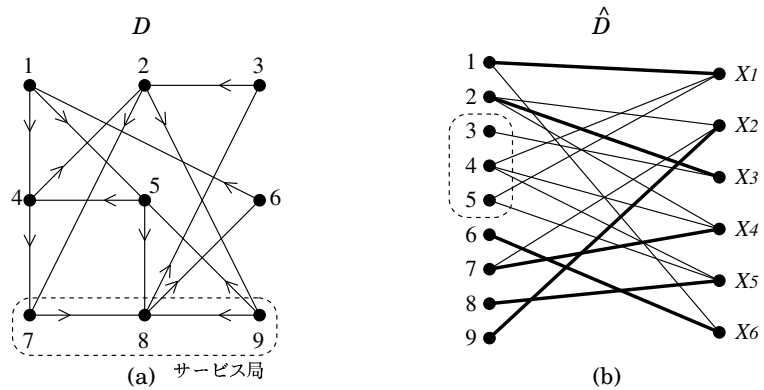


図 2: 有向グラフ D と D に対応する 2 部グラフ \hat{D} .

FMA ではまず，図 2 の (b) のような 2 部グラフ \hat{D} を作る．作り方は，左の頂点集合を基地局 i の集合とし，右の頂点集合をサービス局以外の基地局のコピー X_i の集合とする． i と X_i の間には無条件で辺を設け，有向グラフ D 内に辺 (i, j) が存在する場合かつそのときに限り 2 部グラフ \hat{D} 内に辺 $\{j, X_i\}$ を設ける．

FMA は次に， \hat{D} のある最大マッチング M をとる．そしてその M でマッチされていない基地局 i それぞれに対し，図 3 のような増加木 T_i を構成し， T_i の根から到達可能な基地局 j (根自身も含まれる) からなる集合 V_i をコサーキットとして抽出する．

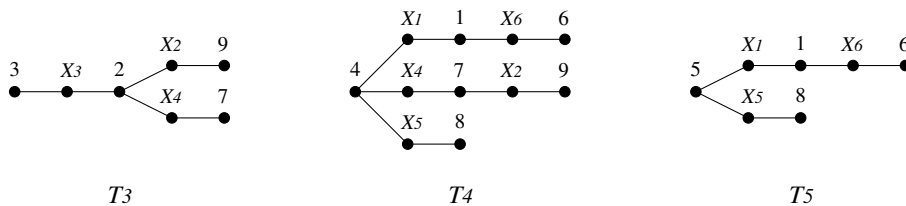


図 3: 増加木.

例えば図 2 の (b) の太線で示すような辺集合をマッチング M としてとると，マッチされていない基地局は $\{3, 4, 5\}$ となる．これらのマッチされていない基地局を根とする増加木 T_3, T_4, T_5 は図 3 のようになり，コサーキットとして抽出される頂点集合は， $V_3 = \{2, 3, 7, 9\}$, $V_4 = \{1, 4, 6, 7, 8, 9\}$, $V_5 = \{1, 5, 6, 8\}$ となる．これにより，表 3 のようなフラット行列が得られ，そこから例えば $P_1 = \{1, 2, 5\}$, $P_2 = \{3, 4\}$, $P_3 = \{6, 7\}$, $P_4 = \{8, 9\}$. のような解が得られる．

	1	2	3	4	5	6	7	8	9
$C(3, B)$	0	1	1	0	0	0	1	0	1
$C(4, B)$	1	0	0	1	0	1	1	1	1
$C(5, B)$	1	0	0	0	1	1	0	1	0

表 3: D に対するフラット行列

本実験で使用したアルゴリズムは上記で説明した FMA とは若干異なり、次の通りである: FMA で分割 D_1, D_2, \dots を求めた後、 $|D_i| \geq |S|/|B|$ なる各 D_i から $|D_i| - (|S|/|B|)$ 個の要素をランダムに抜き取る。次に抜き取られた要素全体の集合は SGA を、それ以外の要素からなる集合は FMA を用いて分割し、最後にその合計を出力する。このように変形された FMA を修正版 FMA と呼び、MFMA で表すことにする。

Strict gammoid の実験に用いたテストデータは次の通り作成した。初めに、500 頂点 (1000 頂点) のランダム無向グラフを辺確率 0.05(0.015) で生成する。そして、辺をランダムに向き付けし、サービス局集合 W を指定された個数だけランダムに選択する。次に、全ての頂点が W のある頂点への有向パスを持つか調べる。そのようなパスを持たない有向グラフは破棄する。破棄されなかったグラフに対して、指定された密度になるまで次の処理を繰り返す: 有向辺 e をランダムに選択し、 e を取り除いても全ての頂点が W のある頂点への有向パスを持つ場合、 e を削除する。表 4 の実験結果より、多くのサービス局を持つ疎な有向グラフに対して、MFMA はよく働くことが分かる。そのような疎な有向グラフに対して、MFMA の結果は SGA とほとんど同じであるが、時々、MFMA は SGA より良い解を作ることがある。しかし、密な有向グラフに対しては、MFMA は SGA に太刀打ちできない。

NS	AD	MFMA	SGA	MFMA	SGA	MFMA	SGA	MFMA	SGA	MFMA	SGA
50	3.6	40	30	36	28	45	45	41	28	35	27
	2.7	41	41	53	48	44	46	52	47	54	53
	2.4	54	59	103	103	107	110	111	110	89	89
100	3.6	23	23	16	16	18	15	17	17	17	17
	2.7	55	55	43	43	48	48	42	42	40	41
	2.4	50	51	53	53	67	68	63	64	65	68
150	3.6	15	15	16	16	19	19	15	15	26	26
	2.7	11	11	11	9	10	10	51	51	38	38
	2.4	57	58	83	82	59	59	50	49	69	71
300	3.6	10	10	10	10	9	9	10	10	12	12
	2.7	20	20	18	18	23	23	24	24	28	28
	2.4	26	26	34	34	44	44	21	21	24	24
450	3.6	8	8	9	9	7	8	8	8	8	8
	2.7	10	10	12	12	36	36	16	16	13	13
	2.4	16	16	14	14	12	12	12	12	11	11
600	3.6	6	7	6	6	5	5	6	6	6	6
	2.7	8	8	10	10	7	7	8	8	10	10
	2.4	11	11	11	11	7	7	7	7	9	9

表 4: 1000 頂点の有向グラフの結果 (NS はサービス局数を表し、AD は平均次数を表す)

謝辞

貴重なご助言を頂いた、国立情報学研究所の宇野毅明先生に感謝いたします。本研究は、日本学術振興会科学研究費補助金 (基盤研究 (C): 課題番号 16500008) の助成を受け行った研究成果の一部である。

参考文献

- [1] S. Arikati, A. Maheshwari and C. Zaroliagis, Efficient Computation of Implicit Representations of Sparse Graphs, *Discrete Applied Mathematics* Vol. 78 (1997) 1-16.
- [2] R.A. Brualdi, Transversal matroids. In N. White, editor, *Combinatorial Geometries*, volume 29 of *Encyclopedia of mathematics and its applications*, chapter 5, Cambridge University Press, 1987. 72-97.
- [3] D. Eppstein, Arboricity and bipartite subgraph listing algorithms, *Information Processing Letters* 51 (1994) 207–211.
- [4] J. Edmonds, Lehman’s switching game and a theorem of Tutte and Nash-Williams, *J. Res. Nat. Bur. Standards Sect. B* 69B (1965) 73-77.
- [5] H.N. Gabow and H.H. Westermann, Forests, Frames, and Games: Algorithms for Matroid Sums and Applications, *Algorithmica* Vol.7 (1992) 465-497.
- [6] A. Itai and M. Rodeh, Finding a Minimum Circuit in a Graph, *SIAM J. Comput.* 7 (1978) 413-423.
- [7] 川野晋一郎, 大館陽太, 山崎浩一, An approximation algorithm for matroid covering, *数理解析研究所講究録* 1426:166-171, 2005.
- [8] J. Oxley, *Matroid Theory*, Oxford University Press, New York (1992).
- [9] E.R. Scheinerman, D.H. Ullman, *Fractional Graph Theory: A Rational Approach to the Theory of Graphs*, Wiley-Interscience Publication. John Wiley & Sons, Inc., New York (1997).
- [10] D. Welsh, *Matroid Theory*, Academic Press, London (1976).