最大フロー手法を応用した論理回路モデルグラフの最小カット列挙法と回路分割手法

畔上 謙吾, 高橋 篤司, 梶谷 洋司

東京工業大学 工学部 電気・電子工学科
〒152-8552 東京都 目黒区 大岡山 2-12-1
Tel : (03)5734-3572, Fax : (03)5734-2902
E-mail : {azegami, atushi, kajitani}@ss.titech.ac.jp

あらまし　最大フロー最小カット定理を応用した，与えられた回路のモデルグラフの最小カットに属す枝を多項式計算時間で列挙し，最小カットを効率良く探索する構造を与えるアルゴリズムについて述べる．本方法の特徴は，無向ハイパーグラフで表現された回路のハイパー枝をフロー対称変換によって有向グラフに変換し，この上で有向グラフに新しく開発した前方探索と呼ぶ頂点探索をくりかえすことにある．この結果を応用し，与えられた回路規模制約のもとで最大回路規模を持つ部分回路を得る手法を示す．これを繰り返し適用することによる回路分割技法を提案する．

キーワード：回路分割，最大フロー最小カット定理

Maxflow based Method for Enumerating Mincut Edges of Graph Modelled Logic Circuit

Kengo Azegami, Atsushi Takahashi and Yoji Kajitani

Dept. of Electrical and Electronic Engrg., Tokyo Inst. of Tech.
Ookayama, Meguro, Tokyo, 152-8552 Japan
Tel : +81-3-5734-3572, Fax : +81-3-5734-2902
E-mail : {azegami, atushi, kajitani}@ss.titech.ac.jp

Abstract　Maxflow mincut theorem based polynomial time computation complexity algorithm for enumerating mincut edges of the graph modelled logic circuits is described. Our algorithm transforms the given undirected hypergraph which shows the target logic circuit into a directed graph. Then, iteratively traverses and clusters the directed graph with our newly developed traversing rule. In this paper, we will give a theorem showing that our algorithm is applicable to any undirected hypergraph when the sources and sinks are given. We will also show an example application in the field of circuit bipartitioning which gives the largest possible subcircuit within the given size constraint.

Key Words : Circuit Partitioning, Maxflow-Mincut theorem

1 はじめに

回路分割は様々な局面で必要なため多くの手法が研究されている（例えば[1] [2]）．その一つとして1955年にFordとFulkersonらが証明した最大フロー最小カット定理に基づく手法がある．この手法は最小カットを多項式時間で見つけるため，回路分割の研究に大きく貢献すると思われる．しかし，フローネットワークの枝は2端子枝（通常の枝）に限られているため，回路が含むファンアウト信号線の扱いが難しい．

ファンアウト信号線は値1でカットされると定義されるべきであるが，回路を適当にモデル化したグラフではその値が複数になる場合を避けられない．ところが文献[3]にファンアウト信号線をカットしてもグラフのカットとしては正しく1本と認識するためのモデル化手法が示されている．我々の回路分割手法もこの文献に示される手法を基本にする．

この手法の別の困難は

1. 分割後の回路規模バランスが予測できない
2. 時間計算量が高次な多項式であるため，大規模問題や，繰り返し計算を行なうアルゴリズムに適さない

などがある．そのため，あまり使われていない．

これらの2つの困難は互いにトレードオフの関係にある．回路規模バランスの調節は，初期解を求め，それを繰り返し修正することで目的の解を求める方法（修正法）で対処している．たとえば，カットのソース側に属す頂点の集まりを逐次変更する方法[3]があるが，そこでは繰り返しフローの計算を行なうので，時間がかかり実用的とは言えない．逐次変更方式である限り，フローの繰り返し計算を回避するためには1回の最大フローの計算結果から，無数にある最小カットから適切なもの選ぶ工夫が必要である．これには少なくとも最小カットに属す信号線のすべてを効率良く見い出す必要があるが，これは一般に困難であるとされている．いずれかの最小カットに属す枝を最小カット枝と呼ぶ．本文はこのすべてを効率的に列挙する問題を解決する．

我々はこの問題の困難は，通常求める最小カットが，カットのいずれかの側の規模が最小である最小カットを求めていることに起因すると考えた．よって，もし1回の最大フローの計算結果からすべての最小カット枝が高速に列挙できれば，最小カットの探索が容易な構造が得られ，最小カットが多項式時間で計算できる利点を享受でき，同時に無数に存在するカットの中から適切なカットを選ぶ手法をとることができるようになると考えた．この推論の妥当性を構成的に示すのが本文の目的である．

2 諸定義

有向グラフGは，頂点の集まりVと枝の集まりEの2項組で示し，$G=(V,E)$と表す．枝は向きを持ち，頂点v_iからv_jに向かう有向枝が存在する時，それを頂点の順序対(v_i,v_j)で表す．このとき，v_jをv_iの後続頂点と呼び，$v_j \in succ(v_i)$と示し，v_iをv_jの先行頂点と呼び，$v_i \in pred(v_j)$と示す．

有向グラフ$G=(V,E)$のカットとは，VをV_1とV_2，ただし$V_1 \cap V_2 = \phi$かつ，$V_1 \cup V_2 = V$なる2つの頂点の集まりに分解する枝の集まり$E_{cut} = E_{12} \cup E_{21}$，ただし$E_{12} = \{(v_1,v_2) \in E \mid v_1 \in V_1$かつ，$v_2 \in V_2\}$，$E_{21} = \{(v_2,v_1) \in E \mid v_1 \in V_1$かつ，$v_2 \in V_2\}$である．カットは$s \in V_1$かつ，$t \in V_2$のとき，$s$-$t$カットと呼ばれ，$E_{12}$に含まれる枝は順方向であるといい，$E_{21}$に含まれる枝は逆方向であるという．

本文で扱うフローグラフは有向グラフの上に定義されており，すべてのフローの源である大ソースと，最終的にすべてのフローが流れ込む大シンクをもち，それぞれsおよびtと表す．フローグラフの枝には，その枝の容量を表す$cap(e_f)$とフローを表す$flow(e_f)$が付与される．$cap(e_f) = flow(e_f)$なら，e_fは飽和しているという．飽和していない枝を非飽和枝，フローが0である枝を0フロー枝と呼ぶ．

フローグラフのs-tカットの容量は，そのカットに含まれる順方向枝の容量の和である．フローグラフの最小s-tカットとは，容量が最小のs-tカットである．

本文では特に断りがない限り，フローグラフとは論理回路を有向グラフでモデル化したグラフとし，最小カットとはフローグラフの最小s-tカットとする．

3 論理回路構造と回路分割

論理回路は論理ゲートや外部入出力端子などを表す回路要素とそれらを接続する信号線の集まりで構成される．

我々の回路分割手法のアウトラインは以下の手順を経る．

1. 論理回路のフローグラフへの変換
2. 最大フローの計算
3. 最小カット枝の列挙
4. 端子数制約を満たし，回路規模が最適な最小カットの選択
5. 選ばれたカットに対応する部分回路の切り出し．
6. 上記操作の回路全体への繰り返し適用．
7. 回路分割の出力．

本手法の鍵は1と3であり，詳細に説明する．

4 跳躍法とその回路分割における問題点

最小カット枝をすべて列挙することは難しいと言われている．最大フロー最小カット定理に基づく場合，その難しさを図1を用いて説明する．

ここでは通常の有向グラフで表されるフローグラフで説明するが，ファンアウトを含む論理回路の分割問題も後述するフロー対称変換によってこのようなフローグラフの最小カット枝列挙問題に帰着されることを言及しておく．

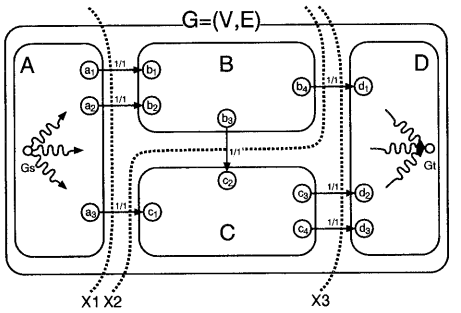

図 1: 最小カットを列挙した図

最大フローを求めたグラフの s にもっとも近い最小カットを見い出すためには s から到達可能な頂点を探索する．到達可能の定義を以下に示す．

定義 1 (到達可能) 頂点 v から到達可能な頂点とは以下のものである．

- v は到達可能である
- v から非飽和枝をその向きに辿って到達できる頂点および，非 0 フロー枝を逆向きに辿って到達できる頂点は到達可能である
- v から到達可能な頂点から到達できる頂点は到達可能である

[定義終り]

グラフの全ての最小カットを X_1, X_2, X_3 とする．そして，A, B, C, D はそれぞれ最小カット枝で囲まれ，かつ，最小カットに横切られない頂点の集まりである．我々は，何らかの手法でこのような構造を見い出したい．

最大フローを流し，s から到達可能な頂点を探索したとする．この時，頂点の集まり A が探索により見い出され，最初の最小カット X_1 を見つけられる．次に $a_1 \in A$ に飽和枝で接続される頂点 b_1 に着目し，そこから新しく探索を再開する根と定める．この操作を（a_1 から b_1 への）跳躍と呼ぶ．そして b_1 から探索する．

この操作で $A \cup B$ が見い出され，次の最小カット X_2 が見つかる．この後，$a_3 \in A$ に飽和枝で接続される頂点 c_1 に跳躍し，同様に探索し，X_3 が見つかる．

すなわち，s, b_1, c_2 または s, b_2, c_2 の順番で到達可能な頂点の集まりを探索することで全ての最小カットを見つけられる．

ここで，s からの探索の後，b_1 ではなく c_1 から次の探索を開始したとする．この時，枝 (b_3, c_2) は定義 1 より探索の対象となり，$\forall b \in B$ において，b は c_1 から到達可能になる．よって，見つけられる最小カットは X_2 ではなく，X_3 になる．b_1 から探索すれば正しく X_2 が見つかるが，そのあと d_1 から探索したら X_3 を正しく見つけられない．

A, B, C, D はそれぞれ回路の一部に相当する．もし A と B を合わせた回路は回路規模の制約を満たすが A, B と C を合わせた回路は満たさない場合，X_2 が見つからなければ A と B を合わせた回路を見つけられず，最適解が得られない．

この A, B, C, D ような，最小カットで分割されない，最小カットで囲まれた部分グラフを見つけるためには，頂点を正しい順番で選び，探索する必要がある．正しい順番を見つけるためには，最初の最小カットを求めた後，その最小カットで切り出される部分グラフに隣接するすべての頂点からの探索を試みる必要がある．その多様性が非多項式オーダである例を作るのは容易である．

5 新手法

我々は最小カット枝で囲まれた部分グラフを効率良く見つけるため，論理回路のモデル化手法と，探索ルールを工夫した．

5.1 モデル化

手続き 1に我々のモデル化手法を示す．本手法は [3] に示される手法に準ずる．論理回路は回路要素および外部入出力端子を頂点，信号線をハイパー枝とした無向ハイパーグラフ $G_c = (V_c, E_c)$ で示されるとする．回路要素 $v_c \in V_c$ に接続される信号線に相当する枝の集まりを $E(v_c)$，論理回路の入出力端子の集まりを $V_t \subset V_c$ とする．

手続き 1 (論理回路のフローグラフ化)

入力

- 論理回路を示すグラフ $G_c = (V_c, E_c)$
- 端子数制約 lim_{io}

出力

- フローグラフ $G_f = (V_f, E_f)$

手続き

1. $\forall e_j \in E_c$ に対し，以下の操作を施す
   (a) e_j に対応する頂点 v_j^i（入力頂点）と v_j^o（出力頂点）を作る
   (b) 容量 1 の有向枝 (v_j^i, v_j^o) を作る
2. $\forall v_j \in V_c$ に対し，以下の操作を施す
   (a) v_j に対応する頂点 v_j^c を作る

(b) $^{\forall}e_k \in E(v_j)$ に対し，以下の操作を施す

i. 容量 ∞ の有向枝 (v_j^c, v_k^i) と (v_k^o, v_j^c) を作る

3. 大ソース s と大シンク t を作る

4. V_t を，V_{t1}, V_{t2}，ただし，$|V_{t1}| \leq lim_{io}/2$ であるように適宜2分割する．

5. $^{\forall}v_i \in V_{t1}$ に対し，容量 ∞ の枝 (s, v_i^c) を作る

6. $^{\forall}v_i \in V_{t2}$ に対し，容量 ∞ の枝 (v_i^c, t) を作る

[手続き終り]

図2に回路例を，図3に，図2に手続き1を施して生成されるフローグラフを示す．論理回路からフローグラフを構成する時間計算量は明らかに回路の規模に対し，多項式である．

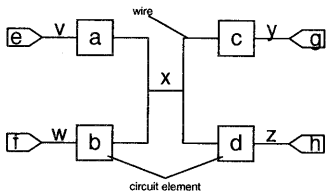

図 2: 回路例

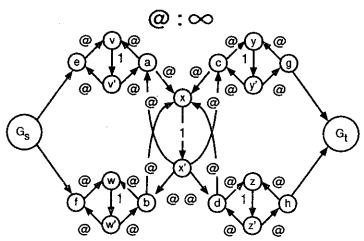

図 3: フローグラフとしてのモデル化

最小カットで切り出される回路は，外部入出力端子として V_{t1} に属す端子に加え，最小カット枝に対応する信号線に接続された端子を持つ．最小カットの容量は，V_{t_1} の数を越えない．よって，$|V_{t1}| < lim_{io}/2$ であるなら，切り出された回路は端子数制約を満たす．回路規模制約を満たす部分回路が見つからない場合は，制約を満たす部分回路が見つかるまで V_{t1} に属す端子を発見的に更新する．

手続き1を用いて構成したフローグラフについて以下の補題が成り立つ．フローグラフの頂点の集まりが，最小カットにより V_{f1}, V_{f2} に分解されたとする．さらに，V_{c1} は対応する頂点が V_{f1} 中にある要素の集まり，V_{c2} は対応する頂点が V_{f2} 中にある要素の集まりとする．この時，最小カットに対応する回路要素の分割を (V_{c1}, V_{c2}) とする．

補題 1 フローグラフの任意の最小カットに対し，カットの容量は，対応する回路要素の分割の間を接続する信号線の数と等しい．

証明:

V_{c1} と V_{c2} を接続する信号線を考える．信号線に対応するフローグラフの容量1の枝が順方向枝として最小カットに含まれないとする．この時，容量 ∞ の順方向枝がカットに含まれ，明らかにそれよりも容量の小さいカットが存在し，矛盾する．よって，信号線に対応して容量1の枝がカットに含まれる．V_{c1} 同士を接続する信号線を考える．信号線に対応する2頂点が V_{f1} に含まれないなら，その2頂点を V_{f1} 側に移動すればカットの容量は減らせ，矛盾する．よって，信号線に対応して容量1の枝はカットに含まれない．V_{c2} 同士を接続する信号線も同様である． ■

5.2 新しい探索ルール

我々は新しい頂点探索ルールを開発し，すべての最小カット枝を多項式時間で見い出した．以下にその新しいルールについて定義する．

定義 2 (前方到達可能) ある頂点 v から前方到達可能な頂点とは以下のものである．

- v は前方到達可能である
- v から非飽和枝をその向きに辿って到達できる頂点は前方到達可能である
- v から前方到達可能な任意の頂点から非飽和枝をその向きに辿って到達できる頂点は前方到達可能である

[定義終り]

以降，“到達可能” といった場合は，従来通りの通常の到達可能を意味することとする．

図3において，頂点 a, b, c, d は回路要素に対応し，それらを接続する信号線に対応する枝は (x, x') である．もし，(x, x') が飽和しているなら，a, b, c, d の任意の2頂点は互いに前方到達可能ではなく，飽和していないなら，互いに前方到達可能である．

回路要素に対応するフローグラフ中の頂点を v_a，v_b，v_c とする．前方到達可能な概念は以下の性質を持つ．

性質 1

- v_b が v_a から前方到達可能なら v_a が v_b から前方到達可能（グラフの構造より）
- v_b が v_a から前方到達可能，v_c が v_b から前方到達可能なら，v_c は v_a から前方到達可能（グラフの構造より）

- 定義2より，v_a は前方到達可能

[性質終り]

前方到達可能関係は，回路要素に対応する頂点の集まりの上で同値関係をなす．よって回路要素に対応する前方到達可能な頂点の集まりは同値類をなす．ただし，信号線に対応する頂点に関しては同値関係は成り立たない．

定義 3 (前方到達可能クラスタ) 最大フローを流したフローグラフの，回路要素に対応する頂点の集まりに対して，性質1で示される同値関係で定義される頂点の集まりを前方到達可能クラスタと呼ぶ． **[定義終り]**

前方到達可能クラスタの集まりを $V = \{V_1, V_2, \cdots V_n\}$ とする．この時，$V_i \cap V_j = \emptyset$ であり $(1 \geq i, j \leq n)$，$\bigcup_{\forall V_i \in V} V_i$ は，回路要素に対応するすべての頂点を含む．前方到達可能クラスタは最大フローを流したフローグラフ中の，回路要素に対応する任意の頂点から前方到達可能な頂点を探索することで見い出せる．フローグラフ中の前方到達可能クラスタを求める操作は探索を始める頂点の順番に依存せず，フローの状態に対して一意に決まる．よって，フローグラフを前方到達可能クラスタの集まりに分解する操作は，フローグラフ中の回路要素に対応する任意の頂点から開始できる．

6 最小カット枝の列挙

前方到達可能クラスタとフローグラフの最小 s-t カットの間には以下の補題が成立する．

補題 2 フローグラフの最小 s-t カットは，前方到達可能クラスタを分割しない．

証明:

前方到達可能クラスタに属す任意の2頂点は，最小カットのいづれかの側に存在することを示す．前方到達可能クラスタに属す任意の2点を v_1，v_2 とする．最大フロー最小カットにおける最小カットの定義より，最大フローを流した時，最小 s-t カットに属す順方向の枝は飽和枝だけである．この時，v_1 と v_2 が最小 s-t カットの s 側と t 側にわかれるなら，定義2より最小 s-t カットに順方向非飽和枝が含まれることになり，最小カットの定義に反する． ■

従って，前方到達可能クラスタの間にある全ての飽和枝の集まりは，すべての最小カット枝を含む．しかし，全ての飽和枝が最小カット枝ではないため，この集まりから最小カット枝の集まりを抽出する操作が必要である．以降，その操作について説明する．

この操作は前方到達可能クラスタ間のフローの関係を用いる．このことを図4を用いて説明する．図は，前方到達可能クラスタ同士がつながっている箇所の一部で，枝 (x, x') に対応する信号線は，a, b, c, d に対応する回路要素を接続する．a, b はクラスタ X_a に，c はクラスタ X_b に，d はクラスタ X_c に属す．フローは a から d に向かって流れている（枝 (a, x)，(x, x')，(x', d) のフローは1，それ以外はすべて0）．

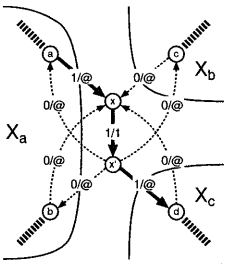

図 4: 飽和枝とその周辺

フローは X_a から流れ出て，X_c に流れ込む．X_b はフローのやりとりに関係ない．フローが流れ出す前方到達可能クラスタとフローが流れ込む前方到達可能クラスタの数は，1本の飽和枝につきちょうど1つづつある．

この関係を元に，最大フローを求めたフローグラフに対し手続き2を適用し新しくグラフを作成する．作成されるグラフを最小カット列挙グラフと呼ぶ．

手続き 2 (最小カット列挙グラフ作成手法)

入力

- フローグラフ $G_f = (V_f, E_f)$

出力

- 最小カット列挙グラフ $G_e = (V_e, E_e)$

手続き

1. G_f のすべての前方到達可能クラスタを求める
2. 求めたそれぞれの前方到達可能クラスタに対応する頂点を作る
3. G_f において異なる前方到達可能クラスタに属す頂点を接続する飽和枝に着目する．この飽和枝は前方到達可能クラスタ X_a，X_b，X_c，$\cdots$ とつながっているとする．フローは X_a に属す頂点から X_b に属す頂点に向かってながれているとする．これに対して次の操作を適用する
   (a) つながっている前方到達可能クラスタの数がちょうど2なら，X_a を表す頂点から X_b を表す頂点に向けて有向枝を作る
   (b) つながっている前方到達可能クラスタの数が3以上なら，X_a を表す頂点から X_c，$\cdots$ を表すそれぞれの頂点に向かう有向枝を作り，X_c，$\cdots$ を表すそれぞれの頂点から X_b を表す頂点に向かう有向枝を作る．

以上の操作で得られたグラフを前方到達可能クラスタグラフと呼ぶ．

4. 前方到達可能クラスタグラフの強連結性分を最小カット列挙クラスタとし，それぞれのクラスタ内の頂点を1点に縮退する．

5. G_e を出力する．

[手続き終り]

図5は，図4に手続き2を施して得られた最小カット列挙グラフを示す．

手続き2は，最小カット列挙グラフに有向枝を，前方到達可能クラスタ間のフローのやりとりに応じて付け加える．前方到達可能クラスタ間のフローのやりとりは飽和枝，入力頂点，出力頂点とそれに接続する容量 ∞ の枝の組合せで示していたが，この手続きで作成される最小カット列挙グラフではこれらの情報を最小カット列挙クラスタ間を結ぶ有向枝として示す．

最小カット列挙グラフの任意のカットには，フローグラフにおいて対応するカットが必ず存在することは明らかである．

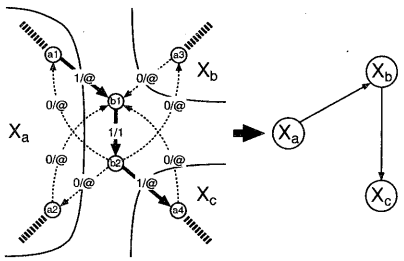

図5: 最小カット列挙クラスタと最小カット列挙グラフ

フローグラフと最小カット列挙グラフに関し，以下の補題が成り立つ．

補題 3 フローグラフの最小 s-t カットは，最小カット列挙クラスタを分割しない．

証明:

ある最小カット列挙クラスタに対応するフローグラフの任意の2頂点は，フローグラフの最小 s-t カットのいずれかの側に存在することを示す．最大フロー最小カットにおける最小カットの定義より，最小カットはフローがある有向枝の集まりである．補題2より，フローグラフの最小 s-t カットは前方到達可能クラスタを分割しない．よって，最小 s-t カットが最小カット列挙クラスタを分割するなら，それは前方到達クラスタの間にある．手続き2より，最小カット列挙クラスタは前方到達可能クラスタグラフの強連結成分を1つのクラスタとして作成する．よって，前方到達可能クラスタグラフにおいて少なくとも1本の有向枝が t 側から s 側にある．t 側にある X_b から s 側にある X_a に有向枝があるとする．この有向枝が手続き2-3-(a)において作られたならば，フローグラフにおいて t から s 側にフローがながれることになり，矛盾する．この有向枝が手続き2-3-(b)において作られたとする．このとき X_a, X_b, X_c に対応する回路要素に接続する信号線が存在し，X_c から X_a へフローがながれているか，または X_b から X_c へフローがながれている．X_c から X_a へフローがながれている場合を考える．X_c が t 側にあるとすると，フローグラフにおいて t から s にフローがながれることになり，矛盾するため，X_c は s 側にある．よって，信号線に対応する点は s 側にあるが，信号線に対応する頂点から X_b に向かう容量 ∞ の枝が最小カットに含まれることになり，矛盾する．X_b から X_c にフローがながれている場合も同様である． ■

最小カット列挙クラスタはフローグラフの任意の最小 s-t カットを考えた時，すべて s 側に属す頂点からなるか，t 側に属す頂点からなる．フローグラフの最小 s-t カットに対応する最小カット列挙グラフのカットは，s 側に属す頂点からなる最小カット列挙クラスタと t 側に属す最小カット列挙クラスタに分割する．

補題 4 フローグラフの最小 s-t カットに対応する最小カット列挙グラフのカットは，有向カットである．

証明:

最小カット列挙グラフのカットが逆方向枝を含むとすると，フローグラフにおいて t 側から s 側へフローがながれることになり，矛盾する． ■

よって，補題3および，4より，フローグラフの最小 s-t カットは最小カット列挙グラフの上で有向 s-t カットである．また，次の補題も成り立つ．

補題 5 最小カット列挙グラフの上の任意の有向カットは，フローグラフの最小 s-t カットに対応する．

証明:

最小カット列挙グラフ上の有向カットは，フローグラフの s-t カットに対応する．もし，対応するフローグラフの s-t カットに順方向非飽和枝が含まれるなら，定義2より，カットの両側に同一前方到達可能クラスタに属す回路要素に対応する点を持つ．よって，有向カットであることに矛盾する． ■

よって補題3および，5より，最小カット列挙グラフ上の全ての有向 s-t カットはフローグラフの全ての最小 s-t カットに対応する．最小カットの数は回路要素数の指数となる場合があるが，最小カット枝は最小カット列挙グラフの枝に対応し，高々多項式本しか存在しない．よって，以下の定理が証明された．

定理 1 (フローグラフの最小カット枝の列挙) フローグラフのすべての最小カット枝は多項式時間で列挙できる． **[定理終り]**

フローグラフ上の最小 s-t カットは論理回路の最小カット，すなわち，ハイパーエッジを持つ無向グラフの最小カットと一致する．よって，以下の定理が証明された．

定理 2 (無向グラフの最小カット枝の列挙) ハイパーエッジを持つ無向グラフの最小カット枝は，始点と終点が与えられる限り，多項式時間で列挙できる． **[定理終り]**

7 最適最小カット

最小カット列挙クラスタを 1 つの頂点とみなせば，最小カット列挙グラフは有向無閉路グラフである．

有向無閉路グラフに関わらず，グラフの最小カットは無数にある．我々が求めたい最小カット列挙グラフ上の最小カットは，s を含み，かつ，切り出す部分回路の端子数が端子数制約を越えない，最大規模の部分回路に対応する部分グラフを切り出すカットである．すなわち問題は，次の定義に従うと以下のように示せる．

$G_e = (V_e, E_e)$ において，$v_i \in V_e$ から $v_j \in V_e$ に至る有向パスが存在する時，v_i を v_j の祖先，v_j を v_i の子孫と呼ぶ．v_i の祖先の集まりを $ans(v_i)$，子孫の集まりを $des(v_i)$ としめす．

定義 4 (祖先関係) $v_i, v_j \in V_e$, $v_i \neq v_j$ において，$v_j \in \{ans(v_i) \cup des(v_i)\}$ なら，v_i と v_j は祖先関係にあるといい，$v_j \notin \{ans(v_i) \cup des(v_i)\}$ なら，v_i と v_j は祖先関係に無いという． **[定義終り]**

G_e の任意の有向カットは $\bigcup_{\forall v_k \in V_j} ans(v_k)$ を s 側とする，互いに祖先関係にない頂点の集まり V_j を持つ．回路規模制約を越えない最大の回路に対応する最小カットを最適最小カットと呼ぶ．

[問題]

$\bigcup_{\forall v_k \in V_j} ans(v_k)$ に相当する，論理回路の回路規模が与えられた回路規模の制約を越えない最大となる互いに祖先関係にない，頂点からなる V_j を求めよ．

[問題終り]

この問題を解くには互いに祖先関係にない頂点の集まりをすべて調べなければならない可能性がある．すなわち，最小カット枝を列挙することは多項式時間で実現できるが，その中から最適なものを選ぶ操作は多項式時間ではできない．そこで探索範囲を狭める工夫を行なう．我々は明らかに回路規模制約を満たさない V_j を除く．以下にその工夫について図 6をもとに述べる．最小カット列挙クラスタを 1 つの頂点として示したグラフを示す．

今，$s \in v_A$, $t \in v_N$ であるとする．

今，最小カット列挙グラフのある最小カット列挙クラスタ v および，v のすべての祖先で構成された集まりを

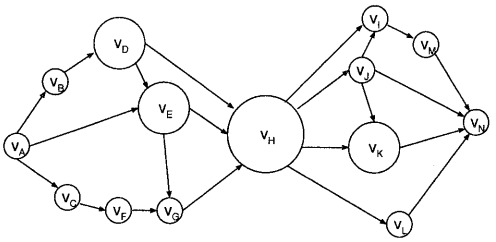

図 6: 最小カット列挙グラフ

考える．この集まりに相当する論理回路が回路規模の制約を越えているとする．この時，v より t 側にある有向カットで最小カット列挙グラフを分割しても，必ず回路規模の制約を越える．

例えば図 6中の v_J および v_J のすべての祖先で構成される集まりに相当する論理回路は回路規模の制約を満たさなければ，v_J の子孫を含めた場合，制約に違反するする．そのため v_J より t 側の点を含む V_j の探索は必要ない．すなわち，最適最小カットは図 7の斜線部分を通ることはない．

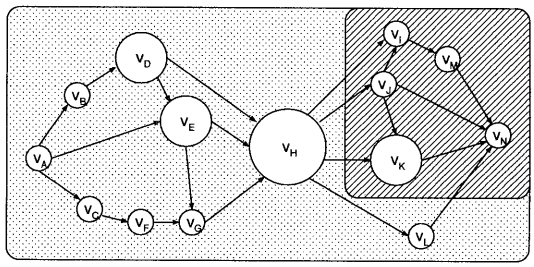

図 7: 最小カット列挙グラフの解を持たない領域

この性質を，最小カット列挙グラフの最適最小カットを全探索する時の問題規模縮小に用いる．手続き 3にその手法を示す．探索対象は最小カット列挙グラフであるとする．

手続き 3 (最大化処理概略)

入力

- 論理回路 G_c
- 最小カット列挙グラフ $G_e = (V_e, E_e)$
- 回路規模制約 lim_{size}

出力

- 回路規模最大の端子数制約を満たす G_c の部分回路

手続き

1. V_e のすべての印の状態を退避する

2. V_e のすべての印を消去する

3. V_e に記録された数値の値を退避する

4. すべての印の付いていない V_e の要素 v_e において，v_e および印のついていないすべての $ans(ve)$ の集まりで構成される部分グラフに対応する論理回路の回路規模を計算し，結果を v_e に記録する

5. V_e のすべての要素 v_e において，記録された回路規模が lim_{size} より大きいなら，$des(v_e)$ のすべては探索不要の印をつける

6. 探索不要の印を元に，解の探索が必要な領域を見つける．解の探索が必要な領域に属すクラスタ v_e の集まりを $X_c \subset V_e$ とする

7. X_c に含まれるすべてのクラスタを一覧表 L に記す．

8. それぞれのクラスタに記録された数値を元に一覧表 L を降順に並べ替える

9. $c_l \leftarrow$ first item in L

10. $L \leftarrow L \setminus \{c_l\}$

11. $L = \phi$ なら印の状態および L を復帰して戻る（再帰の最後のループならステップ24に分岐する）

12. X_l を退避する

13. $X_l = X_l \cup \{c_l\}$

14. すべての印の状態を退避する

15. c_l および，c_l のすべての祖先に印を付ける

16. L を退避する

17. 印のついていないすべてのクラスタを一覧表 L にする

18. L が空でないなら

    (a) すべてのクラスタに記録された数値の値を退避する

    (b) L のすべての要素において，自分自身および印のついていないすべての祖先のクラスタの集まりで構成される論理回路の回路規模を計算し，結果をそのクラスタに記録する

    (c) ステップ8を再帰的に呼び出す

    (d) すべてのクラスタに記録された数値の値を復帰する

19. L が空なら

    (a) X_l のすべての要素におよび，それらのすべての祖先で構成されるクラスタの集まりに相当する論理回路の回路規模を計算する．結果を $size$ とする

    (b) もし，$size \leq lim_{size}$ かつ，前回の $size$ よりも大きいなら，X_l を保存する

20. L を復帰する

21. すべての印の状態を復帰する

22. X_l を復帰する

23. ステップ9に分岐する

24. X_l に相当する部分回路を抽出する

[手続き終り]

この手続きにより，最小カット列挙グラフの有向カットの s 側に相当する論理回路の回路規模が，与えられた回路規模制約を越えない最適な有向カット，すなわち，カットで切り出される論理回路の回路規模が制約条件下でもっとも大きい最小カットを与える．上記手順はクラスタの数に対し，指数時間で必ず終了する．

8 まとめ

本稿では論理回路に最大フロー最小カットを応用し，その全ての最小カット枝を多項式時間で列挙する手法を示した．また，列挙された最小カット枝の集まりから最適なものを求める手法を提案した．フローグラフから最小カット列挙グラフを作成する過程の途中にいくつかのグラフを作成したが，処理高速化のため，一つの手続きとして併合しても差し支えないことは言うまでもない．

9 謝辞

本研究を進めるに当たり御助言頂いた株式会社富士通研究所システムLSI開発研究所第2開発部高橋主任研，マルチメディア研究所CAD研究部河村部長並びに関係者の皆様に感謝致します．本研究はCAD21プロジェクトの一部である．

参考文献

[1] C. M. Fiduccia and R. M. Mattheyses, "A Linear Time Heuristic for Improving Network Partitions', Proceedings of the ACM/IEEE DAC 1982, p. 175

[2] Yen-Chuen Wei, Chung-Kuan Cheng, "Ratio Cut Partitioning for Hierarchical Designs", IEEE Transaction on Computer-Aided Design, vol. 10, No. 7, July 1991, p. 911

[3] Honghua Yang and D. F. Wong, "Efficient Network Flow Based Min-Cut Balanced Partitioning", Proceedings of the ACM/IEEE DAC 1994, p. 50