7A-01

グラフ列挙における構築途中の ZDD 幅に基づく変数順序づけ法

鈴木　浩史 †　　　　湊　真一 †

† 北海道大学　大学院情報科学研究科

1　はじめに

組合せ集合を効率良く圧縮して索引化するデータ構造として Zero-suppressed Binary Decision Diagram (ZDD) [1] が知られている．Knuth は ZDD を用いて，グラフと 2 頂点 s, t が与えられたとき全 s-t パスを索引化するアルゴリズム Simpath を提案した．

これまで，Simpath を拡張した様々なアルゴリズムが開発され，サイクル，全域森，グラフ分割など種々のトポロジー制約を扱えるようになった．これらのアルゴリズムは総称してフロンティア法 [2] と呼ばれている．フロンティア法は，リンクパズルの解法，配電網の最適化，選挙区割の解析など理論的にも社会的にも幅広く応用が行われている [3].

フロンティア法の課題は構築される ZDD の大きさを抑えることであり，それは辺順序に依存して大きく変化することが知られている．従来は，理論的な保証に基づいて，前処理で良い辺順序を見つけるヒューリスティクスが用いられてきた (例えば [4])．しかし，実際の ZDD の大きさは構築が進むまで不明であり，理論的に良い順序でも膨れ上がる場合が多々ある．本研究では，構築途中の ZDD の大きさ，特に幅に着目して，構築と並行して良い辺順序を求める手法を提案する．

2　ZDD とフロンティア法

2.1　ZDD

ZDD は場合分け二分木を簡約化して得られる，組合せ集合のグラフ表現である．組合せ集合とは，台集合 U に対して組合せ $X \subseteq U$ を要素とするような集合である．例えば，$U = \{a, b, c\}$ に対して $\{\{a, b\}, \{b\}, \{a, c\}, \{a, b, c\}\}$ は組合せ集合である．

ZDD はちょうど 1 つの根節点 ρ と 2 つの終端節点 $\bot, \top$ を持つ．終端以外の節点はアイテムのラベルとちょうど 2 つの枝 0-/1-枝を持ち，各枝が指す節点はそれぞれ 0-/1-子と呼ばれる．ここで，アイテムのラベルは根から終端に向かって順序付けられており，子は親のラベルより後のラベルを持つ．これらは非巡回有向グラフを成し，どのようなパスも必ず終端にたどり着く．このとき，ρ から $\top$ へのパス 1 つ 1 つが，ZDD が索引付ける組合せに 1 対 1 対応する．あるパスが表現する組合せは，そのパスが節点 α の 1-枝を含むならば α のラベルを含む，というルールで決定される．(図 1)

2.2　フロンティア法

頂点集合 V と順序付きの辺集合 $E = \{e_1, \ldots, e_m\}$ からなるグラフを $G = \langle V, E \rangle$ と表記する．フロンティア法はグラフ G とトポロジー制約が与えられたとき，制約

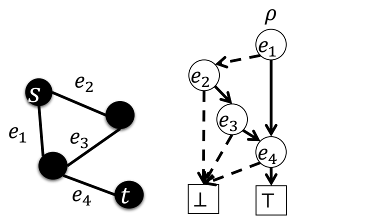

図 1　グラフ (左) から全 s-t パスを索引化した ZDD (右) の例 (破線/実線はそれぞれ 0-/1-枝)

を満たす辺の組合せ $X \subseteq E$ をすべて索引化する ZDD を構築する．ZDD の構築はトップダウンに行われ，根から終端に向かって E の要素をラベルに持つ ZDD が生成される．すなわち，ラベル e_i の節点集合を N_i に格納するとしたとき，$N_1 = \{\rho\}$ および $\bot, \top$ の生成に始まり次の操作を $i = 1, \ldots, m$ について行う．各 $\alpha \in N_i$ と $x \in \{0, 1\}$ について x-子を生成し N_{i+1} に格納するか枝刈りにより $\bot/\top$ に変換する．

フロンティア法で重要になるのは，各節点が管理するデータ configuration である．これは，根からその節点へのパスで決定される辺の組合せによるトポロジーの抽象表現である．2 つの節点 $\alpha, \beta \in N_i$ について，それぞれの configuration が等しければ α を β で置き換えて共有することで ZDD の節点数を削減できる．また，効率的な枝刈りにも利用される．

i ステップ目の処理済み辺 $\{e_1, \ldots, e_{i-1}\}$ と未処理辺 $\{e_i, \ldots, e_m\}$ について，それらの境界に属す頂点集合を F_i と書き i 番目のフロンティアと呼ぶ．節点 $\alpha \in N_i$ が管理する configuration は F_i 中の接続関係に基づく．s-t パスの列挙を例に取ると，図 2 のようなパスの断片構造が configuration となる．

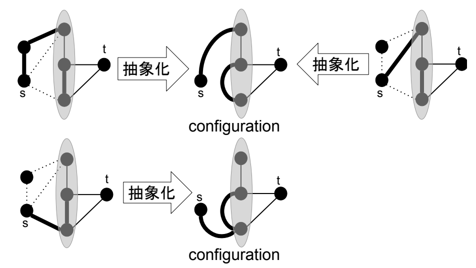

図 2　s-t パスにおける configuration の例 (破線は含まない辺，太い実線は含む辺，それ以外は未処理辺であり，楕円はフロンティアを表す.)

フロンティア法の計算時間とメモリ消費量は各ステップ i で生成する $i+1$ 層目の “ZDD 幅” $|N_{i+1}|$，すなわち管理される configuration の数に依存する．理論的な上界はフロンティアの冪乗で定まる．フロンティアは辺順序から一意に決まるため，その大きさを抑える順序付

A Variable Ordering Method Based on Widths of ZDDs under Construction for Graph Enumeration

†Hirofumi Suzui　†Shin-ichi Minato

†Graduate School of Information Science and Technology, Hokkaido University

けがしばしば用いられる．例えば，井上ら [4] による目先のフロンティアを最小にする辺を選ぶ貪欲法や，ビームサーチという探索法による良い辺順序の探索がある．

3　提案手法

理論的な上界を抑えたとしても，実際の幅が小さくなければ ZDD の構築は困難であり，フロンティアの大きさを抑えることが最良とは限らない．そこで本稿では，あらかじめ与えられた辺順序に従うのではなく，実際の幅が小さくなるように ZDD 構築と辺順序の決定を並行して行う手法を 3 種類提案する．

3.1　手法 1，2

手法 1 は各ステップ $i = 1, \ldots, m$ で，処理済み辺に隣接する未処理辺のうち $|N_{i+1}|$ が最も小さくなる辺を選択する．ここで，最初に処理する辺は必ず e_1 であるとする．これは，フロンティアの大きさに関わらず目先の ZDD 幅を小さくすることを考える貪欲法である．

手法 2 は各ステップ $i = 1, \ldots, m$ で，処理済み辺に隣接する未処理辺のうち $|F_{i+1}|$ が最も小さくなる辺集合に絞り込む．その中で $|N_{i+1}|$ が最も小さくなる辺を選択する．手法 1 と同様に，最初に処理する辺は必ず e_1 であるとする．これは，目先のフロンティアの大きさを優先した上で，ZDD 幅を小さくすることも考える手法である．

手法 1,2 による辺順序は目先の良さのみを考えるため，結果的には良い順序から大きくずれる可能性がある．

3.2　手法 3

手法 3 は手法 1，2 とは異なり，あらかじめ探索された比較的良い辺順序 (例えばビームサーチによる辺順序) を基に，局所的に辺順序を変更する手法である．重要なアイデアは，次のルールに従っていくつかの辺の順序を固定することにある．

- 元の辺順序において，ある辺を処理した後にフロンティアの頂点集合が変化しないならば，その辺を処理する順番は変更しない．

これにより，順序が固定された区間とそうでない区間に分かれる．順序が固定されていない区間に対しては，手法 1 を適用することで局所的に順序の変更を行う．

これは「フロンティアが変化しないならば ZDD 幅も大きく変化しないだろう」という推測に基づく．そして，フロンティアが変化する区間のみに焦点を当て ZDD 幅を抑制することで，元の辺順序の良さを保ちながらより良い辺順序を目指す．これにより，手法 1,2 よりも安定して ZDD 幅の抑制が可能であると考えられる．

4　実験

ビームサーチ [4] に基づく辺順序と 3 つの提案手法で得られる ZDD 幅および計算時間を比較する．各手法であらかじめ与えられる辺順序はビームサーチによるものとした．ここで，ZDD 幅については最大値 $\max\{|N_1|, \ldots, |N_m|\}$ のみを観測する．利用したグラフはグリッドグラフ (8×8 から 10×10)，日本の県隣接関係グラフ (47 頂点 104 辺)，アメリカの州隣接関係グラフ (48 頂点 103 辺)，および watts strogatz モデルによる 40 頂点のランダムグラフ 5 つである．実験結果を表 1 に示す．ZDD 幅について，手法 1 から 3 で従来よりも悪化していない場合に，従来との比を表記している．

表 1 より，手法 3 はいずれのデータに対しても従来よりも悪化していない．しかし，改善のあったデータでは手法 1,2 に比べると改善率はやや劣っている．一方で，手法 1,2 は従来より大きく悪化する場合があり安定性に欠けている．また，手法 1,2 の計算時間が他の 100 倍以上かかる事例があることがわかる．一方で手法 3 はすべてのデータで従来の数倍程度の計算時間で済んでいる．これらの結果から，手法 3 による改善を試みることが妥当な選択になりうると考えられる．

表 1　生成された ZDD 幅と計算時間 (秒) の比較

	ビーム	手法 1	手法 2	手法 3
G8	904	3350	3350	904(**100%**)
	0.230	4.761	4.723	0.398
G9	2377	12029	12029	2377(**100%**)
	0.943	36.973	37.739	2.258
G10	6316	43484	43484	6316(**100%**)
	4.410	212.229	199.265	10.355
Japan	109	90(**82%**)	90(**82%**)	95(**87%**)
	0.005	0.036	0.032	0.007
US	391	306(**78%**)	306(**78%**)	321(**82%**)
	0.015	0.180	0.161	0.030
watts1	15470	9621(**62%**)	9548(**61%**)	12545(**81%**)
	0.563	11.950	9.705	1.506
watts2	2518	40708	3395	2518(**100%**)
	0.087	108.309	4.470	0.130
watts3	2010	2176	2514	2010(**100%**)
	0.067	1.211	1.389	0.133
watts4	2933	2459(**83%**)	2459(**83%**)	2554(**87%**)
	0.112	1.094	1.093	0.191
watts5	1245	6116	6168	1221(**98%**)
	0.056	13.260	10.424	0.174

5　まとめ

本稿では，フロンティア法による ZDD 構築における，構築途中の ZDD 幅を考慮した辺順序づけ手法を 3 つ提案した．実験により手法 3 が妥当な選択になりうるという結果を得た．今後は，各手法の得意不得意を分析し，より安定して改善できる手法の開発を目指したい．

謝辞

本研究の一部は科研費基盤 (S) 15H05711 の助成による．また，助言いただいた本研究室の岡崎文哉さんに感謝致します．

参考文献

[1] Shin-ichi Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. *Proc. of 30th ACM/IEEE Design Automation Conf. (DAC 1993)*, pp. 272–277, 1993.

[2] Jun Kawahara, Takeru Inoue, Hiroaki Iwashita, and Shin-ichi Minato. Frontier-based search for enumerating all constrained subgraphs with compressed representation. *IEICE Trans. Fundamentals*, Vol. E100-A, No. 9, 2017.

[3] ERATO 湊離散構造処理系プロジェクト．超高速グラフ列挙アルゴリズム：「フカシギの数え方」が拓く，組合せ問題への新アプローチ．森北出版，東京，Japan, 4 2015.

[4] Yuma Inoue and Shin-ichi Minato. Acceleration of ZDD construction for subgraph enumeration via pathwidth optimization. TCS-TR-A-16-80. Hokkaido University, 2016.