

# ゼロサプレス型二分決定グラフの省空間化のための近似と圧縮

数理情報学専攻 48186216 松田 康太郎

指導教員 定兼 邦彦 教授

## 1 はじめに

ゼロサプレス型二分決定グラフ (ZDD) [1] は集合族を圧縮した形で表現するデータ構造である。特に疎な集合族に対しては顕著な圧縮効果があることが知られている。ZDD を用いることで、集合族上の様々な有用な演算が ZDD のサイズの多項式時間でできることが知られている [2]。しかし、大きな集合族を表現する ZDD は必要なメモリサイズが許容できないほど大きくなってしまふことがある。本論文では、近似と圧縮の 2 つの方法で ZDD のサイズを小さくすることを考える。

## 2 ゼロサプレス型二分決定グラフ (ZDD)

ゼロサプレス型二分決定グラフ (Zero-suppressed Binary Decision Diagrams, ZDD) [1] は有限集合族を扱うデータ構造である。以下では ZDD と表記する。ZDD は根付きの有向非巡回グラフ  $G = (V, E)$  で、分岐節点と終端節点の 2 種類の節点からなる。終端節点は 0-終端節点  $\perp$  と 1-終端節点  $\top$  であり、終端節点から出ていく枝は存在しない。分岐節点  $v$  にはラベルの整数  $\ell(v) \in \{1, \dots, c\}$  がついており、有向枝である 0-枝と 1-枝を必ず 1 本ずつ持つ。節点  $v$  の 0-枝の行先、1-枝の行先を  $v_0, v_1$  と表す。終端節点のラベルは  $c+1$  とする。ZDD は層  $L_1, \dots, L_{c+1}$  からなり、各層  $L_i$  はラベル  $i$  ( $i = 1, \dots, c+1$ ) の節点からなる。ZDD  $G$  の節点数を  $|G|$  で表現し、ZDD のサイズと呼ぶ。ZDD  $G$  が表す集合族  $\mathcal{F}_G$  を以下のように定義する。

**定義** (ZDD が表す集合族) ZDD の節点  $v$  に対して、集合族  $\mathcal{F}_v$  を再帰的に以下のように定義する。 $v$  が終端節点の場合、 $v = \top$  ならば  $\mathcal{F}_v = \{\emptyset\}$ ,  $v = \perp$  ならば  $\mathcal{F}_v = \emptyset$  とする。 $v$  が分岐節点の場合、 $\mathcal{F}_v = \{S \cup \{\ell(v)\} \mid S \in \mathcal{F}_{v_1}\} \cup \mathcal{F}_{v_0}$  とする。ZDD の根節点を  $r$  とすると、ZDD 全体が表す集合族は  $\mathcal{F}_G = \mathcal{F}_r$  となる。

## 3 ZDD の近似

本論文では、表現する集合族が本来の集合族に対して偽陽性の要素を許すが、偽陰性は決して許さないような特殊な近似 ZDD を考える。つまり、元の集合族の要素を全て含むような集合族を表現する近似 ZDD を考える。このような種類の ZDD を relaxed ZDD [3] と呼

ぶ。relaxed ZDD は確率的メンバーシップクエリや最適化問題の上界計算やメンバーシップオラクルを用いた集合族のサイズの推定などへの応用が考えられる。

本論文では、relaxed ZDD を構築する新たなアルゴリズムを提案する。より偽陽性の要素が少なく、よりサイズが小さい relaxed ZDD を得ることを目標とする。ZDD が与えられる下で relaxed ZDD を構築する提案手法を説明する。同じ層の節点  $u, v$  に対して、包含関係  $\mathcal{F}_u \subseteq \mathcal{F}_v$  が成り立つならば、 $u$  に指されている枝の行先を全て  $v$  に変えることで、偽陰性を生じずに節点  $u$  を削除することが出来る。この操作を  $u$  を  $v$  に併合すると呼ぶ。併合する節点对を決める必要があるが、最適な節点对の集合を求めるのは NP 困難である。そこで、併合する節点  $u, v$  は包含関係が成り立つ節点对の中で目的関数  $\log \left( \frac{|\mathcal{F}_G| + E_G(u, v)}{|\mathcal{F}_G|} \right) / N(u)$  が最小であるようなものを毎回選んで併合する貪欲法を行う。ただし、 $G$  を毎回の併合の直前の ZDD とし、 $E_G(u, v)$  を  $u$  を  $v$  に併合したときに発生する偽陽性の要素数、 $N(u)$  を  $u$  を削除したときに削除可能となる節点の総数とする。この目的関数は 1 節点あたりの偽陽性の要素の発生率を意味している。包含関係や目的関数の計算に必要な値は動的計画法で計算できる。また、ZDD が単調な集合族を表現している場合、特殊な節点を除いた節点  $x$  に対して  $\mathcal{F}_x \subseteq \mathcal{F}_y$ ,  $\ell(x) = \ell(y)$  を満たす節点  $y$  をより高速かつ省空間で見つけられる。この包含関係を利用した近似手法を ZDD の構築手法である深さ優先型フロンティア法に組み込むことで、集合族の情報が与えられた下で relaxed ZDD を直接構築する手法を提案する。

実際にアルゴリズムを実装して性能を実験的に確認した。ZDD が与えられる下で relaxed ZDD を作る設定では提案手法の他に 3 つの手法と比較を行った。1 つ目は、既存手法の heavy branch subsetting [4] である。2 つ目は、節点を減らす際に毎回包含関係全てからランダムに選んで併合を行う手法である (図 1 では random に対応する)。3 つ目は、最も多く節点を減らすことができるような節点对の併合の内、最も偽陽性の発生数が少ないものを貪欲に選ぶ手法である (図 1 では greedy eraser に対応する)。図 1 は通信ネットワーク "Interoute" のマッチングとなる辺集合族を表現する ZDD に対する実験結果である。提案手法は他の手法に

比べて偽陽性の発生数が少ないことが確認された。

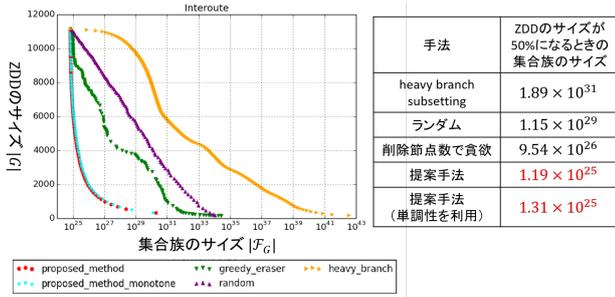


図 1: ZDD の近似に関する実験結果

集合族の情報が与えられる下で relaxed ZDD を作る設定では、提案手法と既存手法の ZDD relaxation [5] の比較を行った。通信ネットワーク "Interoute" のマッチングとなる辺集合族に対する実験結果は図 2 のようになった。提案手法は既存手法よりも偽陽性の発生数が少なくなっていることが確認された。

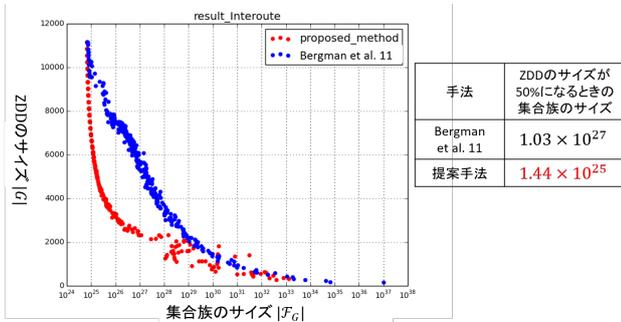


図 2: ZDD の近似に関する実験結果

## 4 ZDD の圧縮

ZDD をラベルがついた特殊な有向非巡回グラフとみなし、圧縮することを考える。木構造の圧縮手法として top DAG 圧縮 [6] が存在する。top DAG 圧縮は木構造内にあるクラスターと呼ばれる部分構造を共有することで圧縮を行うもので、部分木全体が一致していなくても構造の共有を行うことが出来る。また、 $n$  を木の頂点数として、木上の基本的な演算が  $O(\log n)$  時間でできる。top DAG 圧縮を利用することで、ZDD 上の操作が高速に計算できるようにしたまま、繰り返し出てくる部分構造を共有するような省空間化を試みる。

ZDD を全域木とそれ以外の枝 (補木枝) に分け、それぞれに対して省空間で情報を保持するような新しいデータ構造 top ZDD を提案する。top DAG 圧縮を利用することで全域木をより頂点数の少ない DAG として表現する。このとき補木枝を top DAG の適切な頂点に情報を保持することで、辺の情報を共有して持つことができる。top ZDD を用いることで ZDD 上での基本的な演算は  $O(\log^2 n)$  時間で計算可能である。さらに、

データサイズが元の ZDD に対して対数オーダーのサイズになるような ZDD が存在する。また、より省空間化を目指すため、木やビットベクトルに対する簡潔データ構造を利用して必要な情報を圧縮して保持する。

実際に実装し、実験的に性能を確認した。提案手法の top ZDD と既存手法の denseZDD [7] と通常の ZDD の目安となる  $2n[\log n] + n[\log c]$  ビットを比較した ( $n$  を ZDD の節点数,  $c$  を ZDD の層の数とする)。 $A$  個の要素からなる集合  $\{1, \dots, A\}$  のべき集合に対する結果を表 1 に、 $n \times n$  グリッドグラフの左下端点と右上端点を結ぶパスとなる辺集合族を表 2 に示した。これらのケースでは top ZDD は通常の ZDD や denseZDD に比べてデータサイズが小さいことが確認された。

表 1:  $A$  個の要素からなる集合  $\{1, \dots, A\}$  のべき集合に対する実験結果 (単位は bytes)

	top ZDD	denseZDD	$(2n[\log n] + n[\log c])/8$
$A = 1000$	<b>2,297</b>	4,185	3,750
$A = 50000$	<b>2,507</b>	178,764	300,000

表 2:  $n \times n$  グリッドグラフの左下端点と右上端点を結ぶパスとなる辺集合族に対する実験結果 (単位は bytes)

	top ZDD	denseZDD	$(2n[\log n] + n[\log c])/8$
$n = 6$	<b>17,194</b>	28,593	37,441
$n = 7$	<b>49,770</b>	107,529	143,037
$n = 8$	<b>157,103</b>	401,251	569,908
$n = 9$	<b>503,265</b>	1,465,984	2,141,955

## 参考文献

- [1] S. Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. In *Proc. of DAC '93*, pp. 272–277. ACM, 1993.
- [2] D. E. Knuth. *The Art of Computer Programming, Volume 4A : Combinatorial Algorithms, Part 1*. Addison-Wesley Professional, 1st edition, 2011.
- [3] H. R. Andersen, T. Hadzic, J. N. Hooker, and P. Tiedemann. A constraint store based on multi-valued decision diagrams. In *Proc. of CP 2007*, pp. 118–132. Springer, 2007.
- [4] K. Ravi and F. Somenzi. High-density reachability analysis. In *Proc. of ICCAD '95*, pp. 154–158. IEEE Computer Society, 1995.
- [5] D. Bergman, W.-J. van Hove, and J. N. Hooker. Manipulating MDD relaxations for combinatorial optimization. In *Proc. of CPAIOR 2011*, pp. 20–35. Springer, 2011.
- [6] P. Bille, I. L. Gørtz, G. M. Landau, and O. Weimann. Tree compression with top trees. In *Proc. of ICALP 2013*, pp. 160–171. Springer, 2013.
- [7] S. Denzumi, J. Kawahara, K. Tsuda, H. Arimura, S. Minato, and K. Sadakane. Densezdd: A compact and fast index for families of sets. In *Proc. of SEA 2014*, pp. 187–198. Springer, 2014.