

## マトロイド判定問題に対する ZDD を用いた解法とその評価 ZDD-Based Methods for the Matroid Recognition Problem and Their Evaluation

江本 洸海<sup>1)</sup> 岩政 勇仁<sup>1)</sup> 湊 真一<sup>1)</sup>  
Hiromi Emoto Yuni Iwamasa Shin-ichi Minato

### 1 はじめに

マトロイドは、ベクトル空間における線形独立性を抽象化したある性質を満たす有限集合とその部分集合族の組であり、Whitney [1] と Nakasawa [2] によってそれぞれ独立に導入された。部分集合族は課す公理系によって名称が変わり、独立集合族や基族などとよぶ。マトロイドは貪欲法を用いて解ける組合せ最適化問題の背後に潜む数理構造であり、離散構造上の凸性をなす概念である [3]。そのため組合せ最適化分野において重要な役割を果たしているマトロイドが、どのような数理構造から現れるかを考えることは重要である。

本研究では、有限集合  $E$  とその部分集合族  $\mathcal{B} \subseteq 2^E$  の組  $(E, \mathcal{B})$  が何らかの形で与えられたとき、 $(E, \mathcal{B})$  がマトロイドをなすか否か判定する問題 (マトロイド判定問題) について考える。特に  $\mathcal{B}$  が  $E$  上で基族をなすか否かについて考える。この問題に対しては、これまで  $(E, \mathcal{B})$  が公理系を満たすか否か愚直に調べるといった単純な解法や、 $O(|E|^2|\mathcal{B}|)$  時間で判定できる解法が提案されている [4]。基族以外の公理系を前提とした判定も存在する [5]。

しかしマトロイド判定問題を実用的に扱った研究は少ない。 $\mathcal{B}$  のサイズが  $E$  のサイズに対して指数的に大きくなりうる、すなわち入力サイズが大きくなりやすいことが理由の一つである。そこでゼロサプレス型二分決定グラフ (ZDD) によって問題を扱いやすくする。ZDD は Minato [6] によって導入されたデータ構造であり、集合族を圧縮して一意に表現する。

以上を踏まえ、本研究では、マトロイド判定問題に対する ZDD を用いた解法を二つ提案し、計算機実験によって性能を評価する。二つの解法ではそれぞれ ZDD の構造に基づいた再帰関数や、ZDD の特徴である集合演算を主に用いており、ともに ZDD の圧縮効果を活かした解法と言える。マトロイドを判定できれば、さらにマトロイドを前提とした操作や判定を適用できる。たとえばマトロイドの重要な部分クラスであるグラフのマトロイドを判定できる [7]。

### 2 準備

本稿では集合  $X, Y$  に対して、 $\{x \in X : x \notin Y\}$  を  $X - Y$  と表す。単一の要素  $e$  に対して、 $X \cup \{e\}, X - \{e\}$  をそれぞれ  $X \cup e, X - e$  と表す。

#### 2.1 マトロイド

マトロイド [1, 2] はある性質を満たす有限集合とその部分集合族の組であり、具体的には定義 2.1 により定められる。なお記号の用法などは Oxley [8] に従う。

**定義 2.1.** 有限集合  $E$  とその部分集合族  $\mathcal{B} \subseteq 2^E$  が以下をすべて満たすとき、組  $(E, \mathcal{B})$  をマトロイドとよぶ。

1) 京都大学大学院情報学研究所 Graduate School of Informatics, Kyoto University

(B1)  $\mathcal{B} \neq \emptyset$  が成り立つ。

(B2) 任意の  $B_1, B_2 \in \mathcal{B}, e \in B_1$  に対して、 $(B_1 - e) \cup f \in \mathcal{B}$  を満たす  $f \in B_2$  が存在する。

(B3)  $B_1 \subseteq B_2$  を満たす任意の  $B_1, B_2 \in \mathcal{B}$  に対して、 $B_1 = B_2$  が成り立つ。

このとき  $E$  を台集合、 $\mathcal{B}$  を基族とよぶ。

基族には同値な特徴づけが複数存在する。たとえば定義 2.2 によっても基族を定義できる。

**定義 2.2.** 有限集合  $E$  とその部分集合族  $\mathcal{B} \subseteq 2^E$  が (B1) と以下をすべて満たすとき、組  $(E, \mathcal{B})$  をマトロイドとよぶ。

(B4) 任意の  $B_1, B_2 \in \mathcal{B}, x \in B_2 - B_1$  に対して、 $(B_1 - y) \cup x \in \mathcal{B}$  を満たす  $y \in B_1 - B_2$  が存在する。

台集合  $E$  上の基族  $\mathcal{B}$  に対し、 $\mathcal{J} := \{I \subseteq E : \exists B \in \mathcal{B}, I \subseteq B\}$  と定めた  $\mathcal{J}$  を独立集合族とよぶ。実は  $\mathcal{J}$  が有限集合  $E$  上で独立集合族をなすことと、 $E$  の部分集合族  $\mathcal{J} \subseteq 2^E$  が以下をすべて満たすことは同値である。

(I1)  $\emptyset \in \mathcal{J}$  が成り立つ。

(I2) 任意の  $I \in \mathcal{J}, I' \subseteq E$  に対して、 $I' \subseteq I$  ならば  $I' \in \mathcal{J}$  が成り立つ。

(I3) 部分集合  $X \subseteq E$  に対して、 $\mathcal{J}|X := \{I \in \mathcal{J} : I \subseteq X\}$  とおく。任意の  $X \subseteq E$  に対して、半順序集合  $(\mathcal{J}|X, \subseteq)$  における任意の極大元  $I_1, I_2 \in \mathcal{J}|X$  は  $|I_1| = |I_2|$  を満たす。

独立集合族  $\mathcal{J} \subseteq 2^E$  と部分集合  $X \subseteq E$  に対して、 $\mathcal{J}|X$  は  $X$  上で独立集合族をなす。マトロイド  $(X, \mathcal{J}|X)$  を  $X$  への制限とよぶ。半順序集合  $(\mathcal{J}|X, \subseteq)$  の極大元となる集合の要素数を  $X$  のランク  $r(X)$  とよぶ。これは (I3) より一意に定まる。本稿では独立集合族  $\mathcal{J}$  における  $X$  のランクを  $r_{\mathcal{J}}(X)$  と表す。また  $(X, \mathcal{J}|X)$  の基の一つを  $B_X$  とおくと、 $\{I \subseteq E - X : I \cup B_X \in \mathcal{J}\}$  は  $E - X$  上で独立集合族をなす。マトロイド  $(E - X, \{I \subseteq E - X : I \cup B_X \in \mathcal{J}\})$  を  $E - X$  への縮約とよぶ。

無向グラフ  $G$  について、枝集合を  $E(G)$ 、森をなす  $E(G)$  の部分集合の集まりを  $\mathcal{J} \subseteq 2^{E(G)}$  とおくと、 $\mathcal{J}$  は  $E(G)$  上で独立集合族をなす。これをグラフ的マトロイドとよぶ。

半順序集合  $(S, \leq)$ 、部分集合  $S' \subseteq S$  について、任意の  $s \in S, s' \in S'$  に対して  $s \leq s'$  ならば  $s \in S'$  が成り立つとき、 $S'$  を  $(S, \leq)$  の下方集合とよぶ。特に (I2) を満たす  $E$  の部分集合族は半順序集合  $(2^E, \subseteq)$  の下方集合である。本稿では下方集合とよぶとき、べき集合と包含関係についてのものとする。

有限集合上の集合族  $\mathcal{F}$  について、 $F_1 \subseteq F_2$  を満たす任意の  $F_1, F_2 \in \mathcal{F}$  に対して  $F_1 = F_2$  が成り立つとき、 $\mathcal{F}$  を **clutter** とよぶ。(B3) または補題 2.3 より基族は clutter である。

**補題 2.3** ([1, Theorem 6]).  $\mathcal{B}$  を基族とおく。任意の  $B_1, B_2 \in \mathcal{B}$  に対して、 $|B_1| = |B_2|$  が成り立つ。

## 2.2 ゼロサプレス型二分決定グラフ

ゼロサプレス型二分決定グラフ (ZDD) [6] は二分決定グラフの派生形であり、集合族を表す。節点のラベル  $x$  は集合の要素を表し、それぞれ集合に  $x$  を含むか含まないかを表す **1**-枝、**0**-枝という有向枝が各節点に存在する。ZDD  $P$  の 0-枝、1-枝の行き先である ZDD をそれぞれ  $P_0, P_1$  と表す。ただし特別な節点として **0**-終端節点と **1**-終端節点が存在し、これら終端節点にはラベルや 0-枝、1-枝が存在しない。このような節点と有向枝からなるグラフは有向非巡回グラフをなし、根節点から 1-終端節点までの有向パス一つが集合族に属する集合一つに対応する。ZDD  $P$  の根節点のラベルを  $P.top$  と表す。ZDD 内の任意の節点を根とする部分グラフも ZDD となり、これを部分 ZDD とよぶ。

ZDD の特徴はその圧縮規則にある。まず根節点から終端節点までの有向パスに現れる要素の順序を一定にする。すなわち現れる要素を並べた列はすべてある要素列  $x_n, x_{n-1}, \dots, x_1, x_0$  ( $n \in \mathbb{N}$ ) の部分列にする。ここで  $x_0$  は終端節点を表す特別なラベルとして便宜上用意したものであり、集合の要素は表さない。  $i, j \in \{0, 1, \dots, n\}$  に対して、  $x_i > x_j \iff i > j$  と定義する。次に図 1 のように冗長な節点を削除し、図 2 のように等価な節点を共有させることで圧縮する。たとえば図 3 に示す ZDD は集合族  $\{\{x_1\}, \{x_1, x_2\}, \{x_1, x_2, x_3\}\}$  を表す。

本稿では ZDD の節点数を終端節点以外の節点の数とする。表記として集合族と ZDD は同様に扱う。ただし ZDD  $P$  に対して、  $|P|$  は集合族に属する集合の数ではなく、  $P$  の節点数を表す。

ZDD  $P, Q$ 、変数  $x$  に対して、ZDD の集合演算は  $P.change(x) := \{S - x : x \in S \in P\} \sqcup \{S \cup x : x \notin S \in P\}$ ,  $P.onset0(x) := \{S - x : x \in S \in P\}$ ,

$$P \sqcap Q := \{S \cap T : S \in P, T \in Q\},$$

$$P \triangle Q := \{S \in P : \exists T \in Q, T \subseteq S\}$$

などがある。**meet** 演算  $P \sqcap Q$  は  $O(|P|^3|Q|^3)$  時間で計算でき [9]、**restriction** 演算  $P \triangle Q$  は  $O(|P||Q|^2)$  時間または  $O(|P|^2|Q|)$  時間で計算できる [10]。どちらの演算も実用上は高速である。

## 3 マトロイド判定問題

2.1 節で見たように、マトロイドには基族による定義や独立集合族による定義が存在する。ほかにもランク関数などさまざまな概念によるマトロイドの特徴づけが存在する [8]。どの概念を対象とするかによってマトロイド判定問題の難しさは変化する。本稿ではマトロイド判定問題を基族に限定する。

**問題 3.1.** 有限集合  $E$  上の集合族  $\mathcal{B} \subseteq 2^E$  が与えられる。 $\mathcal{B}$  は  $E$  上で基族をなすか。

明示的に有限集合  $E$  が与えられる必要はなく、  $E = \{e \in \mathcal{B} \in \mathcal{B}\}$  とすればよい。

単純な解法として入力  $(E, \mathcal{B})$  がマトロイドの公理系を満たすか否か愚直に調べればよい。マトロイドを扱う数学ソフトウェアではしばしばこの解法を用いている [11, 12, 13]。  $\mathcal{B}$  が非空な clutter という仮定の下で、Provan と Ball は  $O(|E|^3|\mathcal{B}|)$  時間で判定できる解法を提

案した [14]。これを改善する形で Spinrad は  $O(|E|^2|\mathcal{B}|)$  時間で判定できる解法を提案した [4]。

## 4 ZDD を用いた解法

本章では入力の集合族  $\mathcal{B}$  がリストや配列などでなく ZDD の形で与えられたとき、ZDD の圧縮効果を活かして問題 3.1 を解く方法を二つ提案する。集合族  $\mathcal{F}$  を表す ZDD を  $P(\mathcal{F})$  と書くことにし、入力は ZDD  $P(\mathcal{B})$  が与えられるとする。ZDD  $P$  に対して  $E(P) := \{x \in S \in P\}$  とおくと、有限集合  $E$  として  $E(P(\mathcal{B}))$  を考えればよい。

### 4.1 再帰関数を主に用いた解法

本節の解法では補題 4.1 を用いる。補題 2.3 を用いることでこれを示せる。

**補題 4.1.** 有限集合  $E$  上の集合族  $\mathcal{B} \subseteq 2^E$  に対して、  $\mathcal{B}$  が  $E$  上で基族をなす必要十分条件は (B1) と以下をすべて満たすことである。

- (i) 任意の  $B_1, B_2 \in \mathcal{B}$  に対して、  $|B_1| = |B_2|$  が成り立つ。
- (ii)  $\mathcal{J} := \{I \subseteq E : \exists B \in \mathcal{B}, I \subseteq B\}$  とおくと、  $\mathcal{J}$  は (I3) を満たす。

問題 3.1 を解くためには (B1) と (i)-(ii) がすべて成り立つか否か調べればよい。(B1) は定数時間で判定できる。(i) は  $P(\mathcal{B})$  を終端節点から根節点に向かって調べることで、  $P(\mathcal{B})$  の節点数に比例する時間で判定できる。以下では非空な下方集合  $\mathcal{J}$  が (I3) を満たすか否かの判定 CHECKI3( $\mathcal{J}$ ) について考える。

$\mathcal{J} = \{\emptyset\}$  のとき、 (I3) は成り立つ。  $\mathcal{J} \neq \{\emptyset\}$  のとき、部分集合  $X \subseteq E(P(\mathcal{B}))$  が  $P(\mathcal{J}).top \in X$  を満たすか否かで場合分けを行う。ここで  $P(\mathcal{J})_0, P(\mathcal{J})_1$  はともに非空な下方集合であることを示せる。  $P(\mathcal{J}).top \notin X$  のとき、  $P(\mathcal{J}).top$  を含む集合は判定に影響しないため、CHECKI3( $P(\mathcal{J})_0$ ) を調べればよい。  $P(\mathcal{J}).top \in X$  のとき、半順序集合  $S := (P(\mathcal{J})|X, \subseteq), (P(\mathcal{J})_1|X, \subseteq)$  の極大元に注目すると、  $\mathcal{J}$  が (I3) を満たすためには CHECKI3( $P(\mathcal{J})_1$ ) が成り立たなければならないことを示せる。CHECKI3( $P(\mathcal{J})_0$ ), CHECKI3( $P(\mathcal{J})_1$ ) が成り立つとき、それぞれ  $P(\mathcal{J})_0, P(\mathcal{J})_1$  は独立集合族であり、  $\mathcal{J}$  を独立集合族として仮定したときの  $E(P(\mathcal{B})) - P(\mathcal{J}).top$  への制限、または縮約に対応する。このとき  $\mathcal{J}$  が (I3) を満たす、すなわち  $r_{\mathcal{J}}(X)$  が存在することは、  $P(\mathcal{J}).top$  を含まない  $S$  の極大元が存在すれば  $r_{P(\mathcal{J})_0}(X) = r_{P(\mathcal{J})_1}(X) + 1$  が成り立つことと同値である。これも ZDD の根節点に注目した場合分けによって判定できるため、定理 4.2 を示せる。

**定理 4.2.** アルゴリズム 4.1 は非空な下方集合  $\mathcal{J}$  が (I3) を満たすか否か調べる。ただし変数  $e_1, e_2$  に対して、  $\delta_{e_1, e_2}$  はクロネッカーのデルタである。

$P(\mathcal{J})$  の計算には DOWNSET という演算を用いればよい。  $P(\mathcal{J}) = \text{DOWNSET}(P(\mathcal{B}))$  は  $O(|P(\mathcal{B})| \cdot N(P(\mathcal{B}))^2)$  時間で計算できる ( $N(P(\mathcal{B})) := \max\{|\text{DOWNSET}(P')| : P' \text{ は } P(\mathcal{B}) \text{ の部分 ZDD である.}\}$ ) [15]。計算結果を記憶するキャッシュ [6] を用いると、  $O(|P(\mathcal{J})|)$  回 CHECKI3、  $O(|P(\mathcal{J})| \cdot 2^{|E(P(\mathcal{B})|)})$  回 PANDQMATCH を呼び出す。ここで  $\mathcal{B}$  が  $E(P(\mathcal{B}))$  上で基族をなすとき、定理 4.3 より  $|P(\mathcal{B})| = |P(\mathcal{J})|$  が成り立つ。

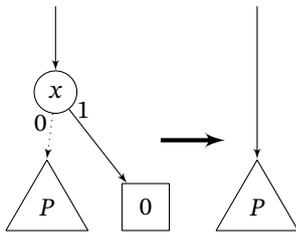


図 1 冗長節点の削除

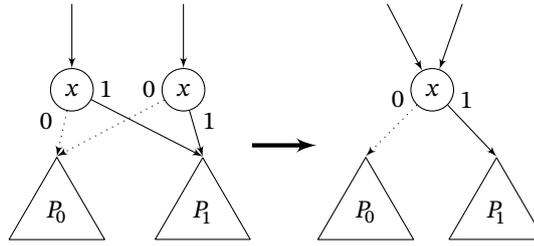


図 2 等価節点の共有

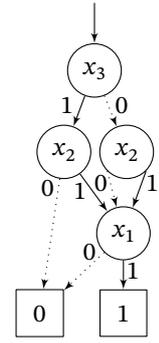


図 3 ZDD の例

**アルゴリズム 4.1** 非空な下方集合に対する (I3) の判定

**Input:** 非空な下方集合  $\mathcal{J}$  を表す ZDD  $P(\mathcal{J})$

**Output:**  $\mathcal{J}$  が (I3) を満たすか。

- 1: **function** CHECKI3( $P(\mathcal{J})$ )
- 2:   **if**  $P(\mathcal{J}) = \{\emptyset\}$  **then return true**
- 3:   **return** CHECKI3( $P(\mathcal{J})_0$ ) **and** CHECKI3( $P(\mathcal{J})_1$ ) **and**  
       PANDQMATCH(1,  $P(\mathcal{J})_0, P(\mathcal{J})_1, P(\mathcal{J})_0 - P(\mathcal{J})_1$ )

**Input:**  $d \in \mathbb{Z}$ , 独立集合族を表す ZDD  $P, Q, R$

**Output:** 任意の  $X \subseteq E(P) \cup E(Q) \cup E(R)$  に対して、  
 $R \setminus X \neq \emptyset$  ならば  $r_P(X) - r_Q(X) = d$  が成り立つか。

- 4: **function** PANDQMATCH( $d, P, Q, R$ )
- 5:   **if**  $R = \emptyset$  **then return true**
- 6:   **if**  $\emptyset \in R$  **then return**  $d = 0$  **and**  $P = Q$
- 7:   **if**  $R.\text{top} > P.\text{top}$  **and**  $R.\text{top} > Q.\text{top}$  **then**
- 8:     **return** PANDQMATCH( $d, P, Q, R_0 \cup R_1$ )
- 9:    $e_{\text{top}} \leftarrow \max\{P.\text{top}, Q.\text{top}, R.\text{top}\}$   
        $\triangleright R \neq \emptyset, \{\emptyset\}$  より  $e_{\text{top}}$  は存在する。
- 10:   **if not** PANDQMATCH( $d, \triangleright e_{\text{top}} \notin X$  のとき  
       **if**  $P.\text{top} = e_{\text{top}}$  **then**  $R_0$  **else**  $P,$   
       **if**  $Q.\text{top} = e_{\text{top}}$  **then**  $Q_0$  **else**  $Q,$   
       **if**  $R.\text{top} = e_{\text{top}}$  **then**  $R_0$  **else**  $R$ ) **then**
- 11:     **return false**
- 12:   **return** PANDQMATCH( $\triangleright e_{\text{top}} \in X$  のとき  
        $d - \delta_{P.\text{top}, e_{\text{top}}} + \delta_{Q.\text{top}, e_{\text{top}}},$   
       **if**  $P.\text{top} = e_{\text{top}}$  **then**  $R_1$  **else**  $P,$   
       **if**  $Q.\text{top} = e_{\text{top}}$  **then**  $Q_1$  **else**  $Q,$   
       **if**  $R.\text{top} = e_{\text{top}}$  **then**  $R_0 \cup R_1$  **else**  $R$ )

**定理 4.3.** 台集合  $E$  上の基族  $\mathcal{B}$  について考える。 $P(\mathcal{B})$  の 0-枝のうち、行き先が 0-末端節点であるもののみ、その行き先が始点の 1-枝と等しくなるようにつけ替える。この操作でできる ZDD  $Q$  は  $\mathcal{B}$  に対応する独立集合族  $\mathcal{J}$  を表し、さらに  $Q$  の節点数と  $P(\mathcal{B})$  の節点数は等しい。

**略証**  $P(\mathcal{B})$  への操作でできるグラフを  $Q'$  とおく。ただし  $Q'$  は圧縮規則の適用前とする。このとき補題 2.3 より  $Q'$  に等価な異なる節点が存在しないことを示せる。そのため  $Q' = Q$  となり、 $|Q| = |P(\mathcal{B})|$  が成り立つ。

集合  $I \in \mathcal{J}$  に対して、 $\mathcal{B}_I := \{B \in \mathcal{B} : I \subseteq B\}$  とおく。自身以外の任意の  $B \in \mathcal{B}_I$  に対して  $\max\{e \in B_I \oplus B\} \notin B_I$  ( $\oplus$  は対称差を表す) を満たすただ一つの集合  $B_I \in \mathcal{B}_I$  に注目し、 $B_I$  を表す  $Q$  上の有向パスを  $D$  とおく。(B4) より  $e \in B_I - I$  を満たす任意の  $e \in E$  に対して、ラベル  $e$  をもつ  $D$  上の節点は 0-枝と 1-枝の行き先が等しいこと

を示せる。そのため  $I$  を表す  $Q$  上の有向パスが存在し、 $P(\mathcal{J}) \subseteq Q$  が成り立つ。またつけ替えた 0-枝には始点と終点ととも等しい 1-枝が存在することに注意すると、 $Q \subseteq P(\mathcal{J})$  を示せるため、 $Q = P(\mathcal{J})$  が成り立つ。  $\square$

**4.2 集合演算を主に用いた解法**

$E := \{e_n, e_{n-1}, \dots, e_1\}$  ( $n \in \mathbb{N}$ ) とおき、有限集合を  $E^+ := \{e_{2n}, e_{2n-1}, \dots, e_{n+1}, e_n, e_{n-1}, \dots, e_1\}$  と拡張する。 $i, j \in \{1, 2, \dots, 2n\}$  に対して、 $e_i > e_j \iff i > j$  と定義する。 $e_{2n}, e_{2n-1}, \dots, e_{n+1}$  を用いて、判定に必要な集合族を ZDD の形で求める。

本節の解法では補題 4.4 を用いる。定義 2.1 と補題 2.3 を用いることでこれを示せる。

**補題 4.4.** 有限集合  $E$  上の集合族  $\mathcal{B} \subseteq 2^E$  に対して、 $\mathcal{B}$  が  $E$  上で基族をなす必要十分条件は (B1)-(B2) と (i) をすべて満たすことである。

問題 3.1 を解くためには (B1)-(B2) と (i) がすべて成り立つかを調べればよい。4.1 節で述べたように (B1) と (i) は比較的容易に判定できる。以下では (B2) の判定について考える。

$\mathcal{R} := \{B - e : e \in B \in \mathcal{B}\}$ ,  $R \in \mathcal{R}$  に対して  $C^*(R) := \{f \in E : R \cup f \in \mathcal{B}\}$ , さらに  $\mathcal{C}^* := \{C^*(R) : R \in \mathcal{R}\}$  とおくと、

$$(B2) \iff \forall B_2 \in \mathcal{B}, \forall C^* \in \mathcal{C}^*, B_2 \cap C^* \neq \emptyset$$

を示せる。 $\mathcal{C}^*$  を表す ZDD  $P(\mathcal{C}^*)$  を計算できれば、 $\emptyset \notin P(\mathcal{B}) \cap P(\mathcal{C}^*)$  が成り立つか否か調べることで (B2) を判定できる。

$f \in \{1, 2, \dots, n\}$  に対して、 $e_f \in C^*(R)$  を満たす  $R \in \mathcal{R}$  を求める。(i) より  $R = B_2 - e_f$  を満たす  $B_2 \in \mathcal{B}$  が存在しなければならぬため、そのような  $R \in \mathcal{R}$  の集合を表す ZDD は  $P(\mathcal{B}).\text{onset0}(e_f) \subseteq P(\mathcal{R})$  となる。

これを踏まえ、 $P(\mathcal{C}^*)$  を求める。初期値を  $P(\mathcal{C}^*) \leftarrow P(\mathcal{R})$  と定める。 $e_f$  と  $e_{f+n}$  を対応づけることで、最終的に  $P(\mathcal{C}^*)$  が  $\{R \cup (E^+ - E) \text{ で表した } C^*(R) : R \in \mathcal{R}\}$  となるように更新する。更新中は常に  $P(\mathcal{C}^*) \cap \{E\} = P(\mathcal{R})$  が成り立つことに注意すると、(i) より定理 4.5 を示せる。

**定理 4.5.** アルゴリズム 4.2 は (i) を満たす  $\mathcal{B}$  が (B2) を満たすか否か調べる。

**5 計算機実験**

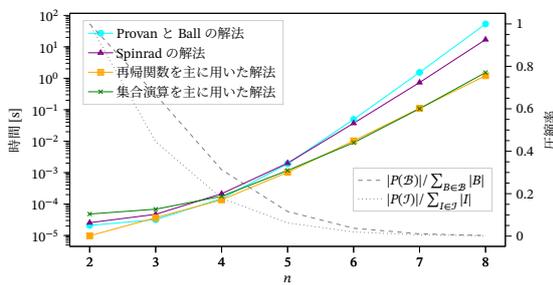
既存手法と提案手法が問題 3.1 の例題を解くまでの時間を測定した。選んだ既存手法は Provan と Ball の解法 [14] と Spinrad の解法 [4] の二つである。すべての解法は C++ で実装し、ZDD のライブラリとして BDD

アルゴリズム 4.2 (i) を満たす  $\mathcal{B}$  に対する (B2) の判定

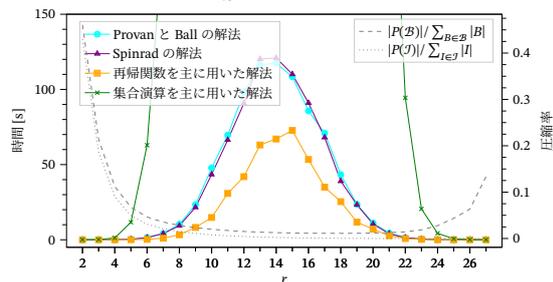
**Input:** (i) を満たす  $\mathcal{B}$  を表す ZDD  $P(\mathcal{B})$

**Output:**  $\mathcal{B}$  が (B2) を満たすか。

- 1: **for**  $i = 1$  **to**  $n$  **do**
- 2:    $P(\mathcal{R}_i) \leftarrow P(\mathcal{B}).\text{onset0}(e_i)$
- 3:  $P(\mathcal{C}^*) \leftarrow \bigcup_{i=1}^n P(\mathcal{R}_i)$    ▷ 初期値は  $P(\mathcal{R})$  に等しい。
- 4: **for**  $f = 1$  **to**  $n$  **do**
- 5:    $Q \leftarrow P(\mathcal{C}^*) \triangle P(\mathcal{R}_f)$
- 6:    $P(\mathcal{C}^*) \leftarrow (P(\mathcal{C}^*) \setminus Q) \cup Q.\text{change}(e_{f+n})$
- 7:  $P(\mathcal{C}^*) \leftarrow P(\mathcal{C}^*) \cap \{\{e_{n+1}, e_{n+2}, \dots, e_{2n}\}\}$   
▷  $\mathcal{C}^*$  を取り出す。
- 8:  $P(\mathcal{C}^*)$  の変数を  $e_{n+i} \rightarrow e_i (i = 1, 2, \dots, n)$  と変換する。
- 9: **return**  $\emptyset \notin P(\mathcal{B}) \cap P(\mathcal{C}^*)$



(a)  $M(K_n)$  を入力とするとき



(b)  $\mathcal{X}_r$  を入力とするとき

図 4 問題 3.1 を解くまでの時間と ZDD の圧縮率

Package SAPPORO-Editon [16] を用いた。実験に用いた計算機の CPU は Intel(R) Xeon(R) CPU E5-2637 v3 @ 3.50GHz、OS は 64 ビットの Ubuntu 16.04.7 LTS であり、1.5 TB のメモリをもつ。

図 4(a) は  $n \in \{2, 3, \dots, 8\}$  頂点完全グラフ  $K_n$  を基にしたグラフ的マトロイド  $M(K_n) = (E, \mathcal{B})$ 、図 4(b) は有限集合  $E$  の要素数を 28 とし、 $|B| = r (r \in \{2, 3, \dots, |E| - 1\})$  を満たす任意の  $B \subseteq E$  に対して  $1/2$  の確率で  $B \in \mathcal{B}$  とする  $\mathcal{X}_r = (E, \mathcal{B})$  を入力としたときの結果である。それぞれにおいて問題 3.1 を解くまでの時間と  $P(\mathcal{B}), P(\mathcal{J})$  の圧縮率を求めた。この例題では再帰関数を主に用いた解法がほかの解法よりもおおむね高速となった。具体的には既存手法より約 2 倍高速であり、ZDD の圧縮効果が大きい  $M(K_n)$  では約 10 倍高速な入力も存在した。同様に集合演算を主に用いた解法も  $M(K_n)$  を入力とする例題では高速であったが、 $\mathcal{X}_r$  では既存手法よりも時間を要した。ほかの解法とは異なり、最後まで計算を打ち切らないことが理由の一つであると考えられる。

## 6 おわりに

本研究では、マトロイド判定問題に対する ZDD を用いた解法を提案し、計算機実験によって既存手法と性能を比較した。提案手法は ZDD の圧縮効果を活かしたものであり、実際に圧縮効果の大きいマトロイドを入力とするときは高速に判定できることを確認した。提案手法のさらなる解析や、マトロイドを表す ZDD 特有の性質の解明が今後の課題である。

### 謝辞

本研究の一部は JSPS 科研費 学変 (A) 20H05964 および基盤 (A) 20H00605 の助成による。

### 参考文献

- [1] Whitney, H.: On the Abstract Properties of Linear Dependence, *American Journal of Mathematics*, Vol. 57, No. 3, pp. 509–533 (1935).
- [2] Nakasawa, T.: Zur Axiomatik der linearen Abhängigkeit. I, *Science Reports of the Tokyo Bunrika Daigaku, Section A*, Vol. 2, No. 43, pp. 235–255 (1935).
- [3] 室田一雄：離散凸解析，共立出版 (2001).
- [4] Spinrad, J.: Note on recognition of matroid systems, *Operations Research Letters*, Vol. 10, No. 6, pp. 313–314 (1991).
- [5] Chaourar, B.: An improved algorithm for recognizing matroids (2019).
- [6] Minato, S.: Zero-Suppressed BDDs for Set Manipulation in Combinatorial Problems, *Proceedings of the 30th International Design Automation Conference, DAC '93*, New York, NY, USA, Association for Computing Machinery, pp. 272–277 (1993).
- [7] Seymour, P. D.: Recognizing graphic matroids, *Combinatorica*, Vol. 1, No. 1, pp. 75–78 (1981).
- [8] Oxley, J.: *Matroid Theory*, Oxford Graduate Texts in Mathematics, Oxford University Press, Oxford, 2 edition (2011).
- [9] Knuth, D.E.: *The Art of Computer Programming, Volume 4, Fascicle 1: Bitwise Tricks & Techniques; Binary Decision Diagrams*, Addison-Wesley Professional, 12th edition (2009).
- [10] Okuno, H. G., Minato, S. and Isozaki, H.: On the properties of combination set operations, *Information Processing Letters*, Vol. 66, No. 4, pp. 195–199 (1998).
- [11] Gawrilow, E. and Joswig, M.: *polymake: a framework for analyzing convex polytopes, Polytopes—combinatorics and computation (Oberwolfach, 1997)*, DMV Sem., Vol. 29, Birkhäuser, Basel, pp. 43–73 (2000).
- [12] Kingan, R. J. and Kingan, S. R.: A software system for matroids, *Graphs and Discovery*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 69, pp. 287–296 (2005).
- [13] The Sage Developers: *SageMath, the Sage Mathematics Software System (Version 9.6)* (2022). <https://www.sagemath.org>.
- [14] Provan, J. S. and Ball, M. O.: Efficient Recognition of Matroid and 2-Monotonic Systems, *Applications of Discrete Mathematics*, Philadelphia, SIAM, pp. 122–134 (1988).
- [15] 菅谷輝治：ZDD を用いた極大集合列挙アルゴリズム，情報処理学会研究報告，Vol. 2014-AL-148, No. 19, pp. 1–8 (2014).
- [16] Minato, S.: BDD Package SAPPORO-Editon (v.20211206) (2021). <https://github.com/Shin-ichi-Minato/SAPPOROBDD>.