

# プレツェルリンクに対する Jones 多項式計算の効率化

内海 友雄\*

今井 桂子†

\* 中央大学大学院 理工学研究科 情報工学専攻

† 中央大学 理工学部 情報工学科

概要： Jones 多項式は結び目の不変量の 1 つであり、Jones 多項式を計算することは  $\#P$ -困難である。本稿では、プレツェルリンクの Jones 多項式を 2 分決定グラフ (Binary Decision Diagrams) を用いて、 $O(n^2)$  時間で計算する方法を示す。プレツェルリンクの中のプレツェルリンクが入っているようなリンクの Jones 多項式を計算する方法も開発した。また、それらに対する計算機実験の結果も示す。

キーワード： 結び目, Jones 多項式, プレツェルリンク, 2 分決定グラフ。

## Computation of the Jones Polynomials for Pretzel Links

Tomoo UTSUMI\*

Keiko IMAI†

\* Information and System Engineering Course,

Graduate School of Science and Engineering, Chuo University

† Department of Information System and Engineering, Chuo University

**Abstract** : In this paper, we consider the problem of computing the Jones polynomials for pretzel links. The Jones polynomial is one of the invariants in knot theory, and it is known that computing the Jones polynomial is  $\#P$ -hard. When we concentrate on the class of pretzel links, we can show that there is an efficient algorithm for that class. We present  $O(n^2)$  algorithm for computing the Jones polynomial of a pretzel link. Moreover, we can use the same approach to a class of links related with pretzel links. Computational results are also reported.

**keyword** : knot, Jones polynomial, pretzel link, binary decision diagrams.

### 1 はじめに

結び目・絡み目といった位相に関する構造を、計算量の観点から研究する分野として、計算位相学と呼ばれる分野が形成されてきている。結び目に関する基本的問題の内、与えられた結び目が自明なものであるかという判定問題については、1999 年に J. Hass, J. C. Lagarias, N. Pippenger によって、この問題が NP の計算量のクラスに属することが示された [2]。

結び目の自明性・同値性判定などと深く関係し、それ自体も研究対象であるものに、結び目の不変量の理論がある。古くから有名なものとして Alexander 多項式がある。これは 1 変数不変多項式で、多項式時間で計算できる。しかし、自明な結び目と同じ Alexander 多項式をもつ自明でない結び目も知られており、一般に結び目の判定能力はそれほど高くない。それに対し、1 変数多項式不変量である Jones 多項式は V.F.J. Jones により定義され、自明でない結び目で自明な結び目と同じ多項式をもつものは、現在まで知られておらず、Jones 多項式は結び目の判別の力が一般に優れている。この多項式の計算法は L.H. Kauffman により結び目のリンク・ダイアグラム  $D$  から計算す

る方法が考案された。しかし、この計算法では、結び目のリンク・ダイアグラムの交点数を  $c(D)$  としたときに、 $2^{c(D)}$  個の多項式の和を計算しなくてはならない。実際、D.J.A. Welsh により Jones 多項式の計算量は  $\#P$ -困難であることが示されており [7]、最悪指数時間かかると思われている。Jones 多項式を計算するアルゴリズムとしては、関根らによる  $O(2^{O(\sqrt{c(D)})})$  時間アルゴリズムが知られている [5]。また、2 つの結び目の同値性判定については有効なアルゴリズムが知られておらず、2 つの結び目の Jones 多項式を計算して、それが違う場合には同値でないという計算証明法が重要となる。

しかし、入力するリンク・ダイアグラムに制限を加えると、多項式時間で計算できる場合があることが知られている。1999 年に Mighton が Tait グラフの tree width が高々 2 であるとき Kauffman bracket 多項式が  $O(c(D)^4)$  時間でできることを示している [4]。また、2000 年に Makowsky は、tree width が定数のとき colored Tutte 多項式を計算する多項式時間アルゴリズムを示し [3]、それを利用することにより、Tait グラフの tree width が定数のとき、その Kauffman bracket 多項式が  $c(D)$

の多項式で計算できることを示している。また、原らが Arborescent link の Jones 多項式  $O(n^3)$  時間で求められることを示した [1]。

本稿では、関根らのアルゴリズム [5] を一般の結び目ではなく、特殊な結び目に用いると Jones 多項式を多項式計算できることを示し、その計算機実験の結果を示す。

## 2 結び目と絡み目

2つの3次元の紐の絡み目が位相的に同じであるかどうかを判定するには相異判定が容易な代数系を見つけて、それらと比較して、判定することが必要となる。これらの数量や代数系を結び目の不変量という。

### 2.1 リンク・ダイアグラム



図 1. 結び目 図 2. リンク・ダイアグラム

結び目とは、円  $S$  を 3 次元 Euclid 空間  $\mathbb{R}^3$  へ埋めこんだものである。絡み目  $L$  とは、結び目の有限個の集まりのことをいう。3 次元の絡み目  $L$  を 2 次元平面に射影したものを考える。平面上に得られた絡み目の像において多重点は 2 重点だけになるように射影する。これをリンク・ダイアグラム  $D(L)$  という (図 2)。

### 2.2 ブラケット多項式

まず、Jones 多項式と関係の深い Kauffman bracket 多項式 (以下 ブラケット多項式) について説明する。ブラケット多項式  $\langle D \rangle$  は与えられた絡み目のリンク・ダイアグラム  $D$  に次の 3 つの規則を適用して得られる 1 変数  $A$  の多項式である。

- (i)  $\langle U \rangle = 1$
- (ii)  $\langle DU \rangle = -(A^2 + A^{-2}) \langle D \rangle$
- (iii)  $\langle \begin{array}{c} \diagup \\ \diagdown \end{array} \rangle = A \langle \begin{array}{c} \diagdown \\ \diagup \end{array} \rangle + A^{-1} \langle \begin{array}{c} \diagup \\ \diagup \end{array} \rangle$   
 $\langle \begin{array}{c} \diagdown \\ \diagdown \end{array} \rangle = A \langle \begin{array}{c} \diagdown \\ \diagdown \end{array} \rangle + A^{-1} \langle \begin{array}{c} \diagup \\ \diagdown \end{array} \rangle$

ただし、 $U$  は自明な結び目、 $DU$  は自明な結び目との union を表すとす。また (iii) はリンク・ダイアグラムの各交点で局所的に適用され、左辺の上下が反転したものについては右边を対応して更新する。

## 3 Jones 多項式

### 3.1 Jones 多項式とブラケット多項式の関係

絡み目のリンク・ダイアグラムにおいて、各結び目で同じ向きになるようにリンク・ダイアグラムに向きを付ける。この向きを付けたリンク・ダ

イアグラムが交差している場所で、2 つの向きが下から上へ行くように見たとき、左下から右上に向かう交差が上にあるときに +、逆ならば - の符号をつける。これを向きのついた絡み目  $L$  と呼ぶ。

向きの付いた絡み目のねじれの数  $w(D)$  とは、向きのついたリンク・ダイアグラム  $D$  における交点の符号 (+1 もしくは -1) の総和で表される。任意の向きのついた絡み目  $L$  のリンク・ダイアグラム  $D$  に対して、Jones 多項式  $V_L$  とブラケット多項式  $\langle D \rangle$  は次式のような関係にある。

$$V_L(A^{-4}) = (-A^3)^{-w(D)} \langle D \rangle$$

以下では、Jones 多項式を求める代わりにブラケット多項式を求める方法を考えていく。

### 3.2 ブラケット多項式の再帰計算式

隣り合うリンク・ダイアグラムの領域が同じ色にならないように白黒の 2 色によって色分けすることができる。このとき、色分けされた領域のうち黒の部分点を  $v$  とする。もし 2 つの黒の領域が交点を境界で共有しているならば、黒の領域を表す点を線分で結びそれを辺  $e$  とし、向きのついた絡み目  $L$  における交点の符号を辺  $e$  につける。このようにして得られるグラフを Tait グラフ  $S$  (以下、グラフは Tait グラフを意味する) と呼ぶ。グラフ  $S$  から辺  $e$  を削除して得られるグラフを  $S \setminus e$ 、 $S$  から辺  $e$  を縮約して得られるグラフを  $S/e$  で表す。両端が同一頂点であるような辺をループ、その辺を削除するとグラフの連結成分数が 1 増えるような辺を橋と呼ぶことにする。

グラフ  $S$  に対して辺の縮約・削除公式が成り立つことが Thislethwaite [6] によって示された。

それらを用いると再帰計算式が得られる。辺  $e$  が正の橋の場合には、 $S \setminus e$  は非連結 (すなわち  $S \setminus e$  に対応する絡み目の成分数が 1 つ増加) となるが、 $S \setminus e$  と  $S/e$  は 2 同形であり、これらの間には次のような関係式が成り立つ。

$$\begin{aligned} \langle S \setminus e \rangle &= (-A^2 - A^{-2}) \langle S/e \rangle \\ \text{従って辺 } e \text{ が正の橋の場合には、} \\ \langle S \rangle &= A \langle S/e \rangle + A^{-1} \langle S \setminus e \rangle \\ &= -A^{-3} \langle S/e \rangle \end{aligned}$$

負の橋の場合も同様であり、まとめると次のような再帰計算式が成り立つ。

$$\langle S \rangle = \begin{cases} -A^{-3} \langle S/e \rangle & e \text{ が正の橋} \\ -A^3 \langle S/e \rangle & e \text{ が負の橋} \\ -A^3 \langle S \setminus e \rangle & e \text{ が正のループ} \\ -A^{-3} \langle S \setminus e \rangle & e \text{ が負のループ} \\ A \langle S/e \rangle + A^{-1} \langle S \setminus e \rangle & e \text{ が正} \\ A \langle S \setminus e \rangle + A^{-1} \langle S/e \rangle & e \text{ が負} \end{cases}$$

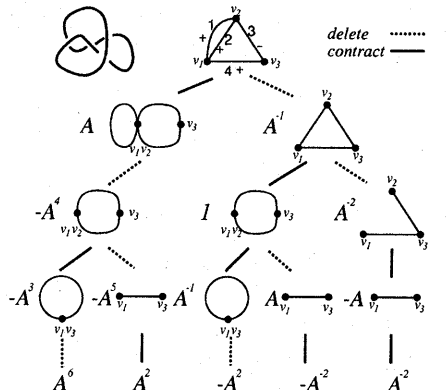


図 3. 再帰計算を用いたブラケット多項式の計算例

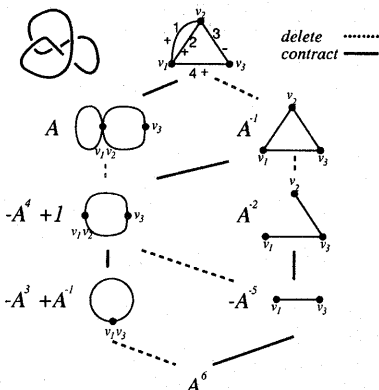


図 4. BDD を使った計算例

### 3.3 2 分決定グラフ

グラフ  $S$  の辺集合  $E$  の辺順序を  $e_1, e_2, \dots, e_n$  ( $|E| = n$ ) とし、レベル  $i$  において辺  $e_i$  を縮約または削除する。このときレベル  $i$  において既に縮約・削除を行った辺集合を  $E_i = \{e_1, e_2, \dots, e_i\}$ 、その補集合を  $\overline{E}_i = \{e_{i+1}, e_{i+2}, \dots, e_n\}$  とする。レベル  $i$  のエリミネーション・フロントとは、 $E_i$  に属する辺および  $\overline{E}_i$  に属する辺の両方に接続する頂点集合のことである。エリミネーション分割とはレベル  $i$  のエリミネーション・フロントに属する頂点において、縮約により統一された頂点を同値類とみなした頂点分割のことである。この時、次の定理が成り立つ。ここでマイナーとは縮約・削除を繰り返して得られるグラフのことである。

**定理 1**  $H_1, H_2$  を辺集合  $\overline{E}_i$  からなる  $S$  のマイナーとする。  $H_1$  と  $H_2$  が辺の順序も含めて同型である必要十分条件は、  $H_1$  と  $H_2$  のエリミネーション分割が等しいことである。

符号付きグラフ  $S$  においては、辺  $e$  の縮約・削除をおこなっても残りのどの辺の符号も変化しない。従って辺の順序も含めて同形なマイナーは、各辺の符号も含めて同形である。上記の定理 1 をもとに、辺の順序も含めて同形なマイナーをすべて統一してブラケット多項式を計算することができる。このように辺の順序も含めて同形なマイナーを統一した計算過程のグラフは、2 分決定グラフ (Binary Decision Diagrams) と関係をもつ。そこで、この計算過程のグラフを単に BDD と呼ぶ [5]。また、各レベルのマイナーの数はエリミネーション・フロントの数のベル数以下になる。

### 4 アルゴリズム

この BDD を用いたブラケット多項式の計算は次の手順でおこなえばよい (図 5)。符号付きグラフ  $S$  の辺に順序を付けておく。  $sign_i$  は、  $e_i$  の辺の符号を  $\{1, -1\}$  で表現したものである。  $\tilde{G}$  は操

作しているマイナー、  $e_i$  は操作している辺、また、  $u, u'$  は  $e_i$  の端点とする。  $f$  はエリミネーション・フロントの数とする。

この手順で計算したときの計算時間を考える。エリミネーション・フロントを見つけるのは、これまでに操作した辺と操作していない辺の両方の辺を持つ頂点を見つけるだけなので  $O(n)$  である。ループの判定は、  $\tilde{G}$  のエリミネーション分割で、  $u, u'$  が同じ集合に入っているかを判定すれば良いので、  $O(f)$  ができる。また、橋の判定は、  $e_i$  を削除したときに連結成分数が 1 増加となるかの判定であるので、  $\tilde{G} \setminus e_i$  において  $u$  から  $u'$  へのパスが存在するかを判定すれば良い。従って、  $O(fn)$  時間である。各マイナー  $\tilde{G}$  における多項式は最大で最高次数  $3n$ 、最低次数  $-3n$  である  $O(n)$  項の多項式を持つことがわかる。よって、その和や単項式との積は  $O(n)$  ができる。また、2 同形なマイナーの探索は、そのレベルでの  $B_f$  個のマイナーとエリミネーション分割が等しいかどうかを調べるので、  $O(fB_f)$  となる。よって、このアルゴリズムは、  $O(n^2 B_f + fn B_f^2)$  となる。一般にベル数  $B_f$  は  $O(f^f)$  である。エリミネーション・フロントの数  $f$  が定数となるような絡み目については  $O(n^2)$  で計算できる。

### 5 プレツェルリンク

前節までで述べたように、一般の絡み目に対して Jones 多項式を求めるには指数時間の計算量が必要となる。しかし、特殊な絡み目なら効率良く計算できると思われる。ここでは、プレツェルリンク (図 6) を考える。

プレツェルリンクは  $P(q_1, q_2, \dots, q_m)$  ( $q_i \in \mathbb{Z}$ ) で表す。また、  $m$  は 3 以上であり、  $|q_k|$  はそれぞれ交差の数、  $q_k$  の符号は交差の符号を表すものである。  $q_k$  中に別符号となるような交差があるとき、同一符号になるように  $q_k$  の部分でねじれの回数を単に増減させるだけで、単一符号となるように

```

Input: 順序の付いた グラフ S
Output: The < D >
 $\tilde{G} := S$ 
 $S_0 := \{\tilde{G}\}$ 
 $T(S_0) := 1$ 
begin
  for  $i := 1$  to  $n$  do
    begin
      エリミネーション・フロントを見つける
       $S_i := 0;$ 
      for  $S_{i-1}$  のそれぞれのマイナー  $\tilde{G}$  do
        begin
          if  $\tilde{G}$  中で  $e_i$  がループ then
             $M(\tilde{G}) := \{\tilde{G} \setminus e_i\}$ 
             $T(\tilde{G} \setminus e_i) := -A^{3 \times \text{sign}_i} \times T(\tilde{G})$ 
          else if  $\tilde{G}$  中で  $e_i$  が橋 then
             $M(\tilde{G}) := \{\tilde{G} / e_i\}$ 
             $T(\tilde{G} / e_i) := -A^{-3 \times \text{sign}_i} \times T(\tilde{G})$ 
          else  $M(\tilde{G}) := \{\tilde{G} / e_i, \tilde{G} \setminus e_i\}$ 
             $T(\tilde{G} / e_i) := A^{1 \times \text{sign}_i} \times T(\tilde{G})$ 
             $T(\tilde{G} \setminus e_i) := A^{-1 \times \text{sign}_i} \times T(\tilde{G})$ 
          for  $M(\tilde{G})$  のそれぞれのマイナー  $G_{e_i}$  do
            begin
              if  $S_i$  中に  $G_{e_i}$  と 2 同形なマイナー  $\tilde{G}$  が
                存在する then
                   $T(\tilde{G}) := T(\tilde{G}) + T(\tilde{G}_{e_i})$ 
              else
                 $S_i$  に  $G_{e_i}$  を追加する
            end
          end
        end
      end
    end
  end
end

```

図 5. 計算手順

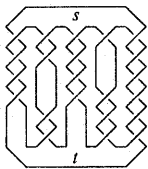


図 6. プレッツェルリンク

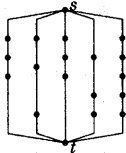


図 7. プレッツェルリンクのグラフ

することができるので、単一符号となるように表現したものを用いる。また、プレッツェルリンクの上側の領域を  $s$ 、下側の領域を  $t$  とする。そのグラフは図 7 のようになる。このようなプレッツェルリンクに対する効率の良い辺順序を考える。

絡み目のリンク・ダイアグラムにおいて、絡み目がちょうど 4 箇所でお互い交わるような円でお互いに囲まれた領域をタングルと呼ぶことにする。ただし、絡み目が円とお互い交わる 4 箇所は、左上 (NW)、右上 (NE)、左下 (SW)、右下 (SE) にあるものとする。NW と NE、SW と SE とを交わらないように結んだタングルを  $\infty$ -タングルと呼ぶ。また、NW と NE を SW と SE と交わらないように結んだタングルを 0-タングルという。0-タングルを  $n$  回ねじって

できるタングルを  $n$ -タングルといい、総称して整数タングルという (図 8)。

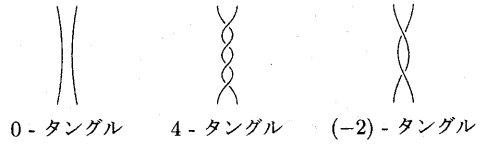


図 8. タングル

この表現をプレッツェルリンクに用いると、 $I_{q_k}$  を整数タングルとした図 9 が得られる。また、プレッツェルリンクのグラフにおいて、タングル  $I_{q_k}$  は  $s$  から  $t$  へ 1 本の道  $p_k$  となる。

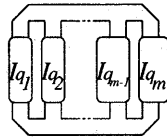


図 9. プレッツェルリンクのタングルでの表現

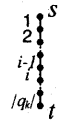


図 10. タングル  $q_k$  のグラフ  $p_k$

このタングルのグラフに  $p_1$  から順に  $s$  から  $t$  へ下向きに辺順序をつける。このとき、 $i$  番目の辺を計算するのに必要なエリミネーション・フロントは、他にエリミネーション・フロントがないとすれば、 $s$  と  $e_i, e_{i-1}$  が共有する頂点だけである。また、 $p_k$  上の辺を全て計算し終えたときにエリミネーション・フロントとして  $s, t$  を持つことになる。このことから、 $p_k$  において上から順に辺順序をつければ、エリミネーション・フロントとして  $s, t, e_i$  と  $e_{i-1}$  の共有点の 3 個の点を持つことになる。 $p_k$  が終わったら  $p_{k+1}$  を計算することにする。この様に辺順序をつけるとプレッツェルリンク全体として常に 3 個以下のエリミネーション・フロントしか持たない。

**定理 2** プレッツェルリンクのグラフが与えられたとき、BDD を用いてプレッツェルリンクの Jones 多項式を計算するのに必要な記憶容量は  $O(n)$  であり、 $O(n^2)$  時間で計算できる。

**証明:** エリミネーション・フロントが定数でおさえられる辺順序を見つけることができるので、各レベルのマイナーの数は定数でおさえることができる。アルゴリズムからエリミネーション・フロントの数が定数ならば、 $O(n^2)$  時間で計算できる。 □

## 6 プレッツェルリンクを拡張したリンク

プレツェルリンクのグラフ  $S$  は  $s$  と  $t$  の間に複数の道を持つようなグラフである。このようなグラフを  $P$  グラフということにする。  $P$  グラフの各道にまた  $P$  グラフを挿入することを考え、挿入した後のグラフも  $P$  グラフ (図 11 の実線) と呼ぶことにする。対応するリンクはプレツェルリンクの中にプレツェルリンクを含んでいるようなリンクと考えられる (図 11 の点線)。内部に  $P$  グラフを含まないとき単純ということにする。  $P$  グラフ  $P_1$  の道に  $P$  グラフ  $P_2$  が含まれている時、  $P_1(P_2)$  と書き、1つの道に  $P$  グラフ  $P_2, P_3$  が直列に含まれている時、  $P_1(P_2|P_3)$  と表す。一方、  $P$  グラフ  $P_1$  の2つの道に  $P$  グラフ  $P_2, P_3$  が含まれている時には、  $P_1(P_2, P_3)$  と表す。これらを組み合わせることによって、様々な形の  $P$  グラフを記述することができる。

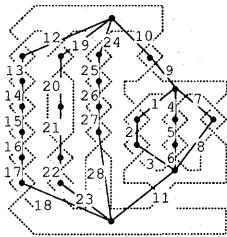


図 11.  $P$  グラフと対応するリンク

1つの道の中に直列に  $P$  グラフが含まれている場合 ( $P_1(P_2|P_3)$  のような形の  $P$  グラフ) は含まれている  $P$  グラフのうち最も上端点に近いもの ( $P_2$  とする) から計算する。このとき、エリミネーション・フロントは  $P_2$  の上端点  $s_2$  と下端点  $t_2$ 、操作する辺の1つの端点の3つである。計算が終わった時、エリミネーション・フロントは2点となり、  $s_2$  から上に向かって  $P_1$  の上端点  $s_1$  に到達するまで辺を操作していくが、この間も2点である。その後、  $t_2$  から下へ辺を操作し、もし他の  $P$  グラフが存在したら、その  $P$  グラフ ( $P_3$  とする) の内部を計算する。このときのエリミネーション・フロントは  $s_1$  と  $P_3$  の  $s_3, t_3$  と操作する辺の1つの端点の4つである。計算を終えると、エリミネーション・フロントは  $s_1$  と  $t_3$  の2点である。これを  $t_1$  に到達するまで繰り返す。また、この計算が終わったとき、エリミネーション・フロントは  $s_1$  と  $t_1$  の2点である。次に  $P_1$  の  $P$  グラフを含んでいる道以外を計算すれば、  $s_1, t_1$  と操作する辺の1つの端点の3点で計算できる。このように、1つの道にしか内部の  $P$  グラフが存在しなければ、全体  $P_1$  としてエリミネーション・フロントは高々4点である。図 11 の番号は辺順序を示したものである。

また、1つの道に再帰的に1つだけ  $P$  グラフが含まれる場合 ( $P_1(P_2(P_3))$ ) のような形の  $P$  グラフ (図 12) は含まれているグラフの一番中のグラフ  $P_3$  から計算する。このとき、  $P_3$  を計算しているとき、エリミネーション・フロントは  $s_3, t_3$  と操作する辺の1つの端点の3つである。計算が終わったとき、  $s_3, t_3$  の2点である。次に、  $s_3$  上に向かってから  $s_2$  まで辺を操作し、その後、  $t_3$  から下へ  $t_2$  まで辺を操作する。この間も2点である。次に  $P_2$  について、  $P_3$  を含んだ道以外の操作は3点で計算できる。また、  $s_2$  から  $s_1, t_2$  から  $t_1$  へも同様に操作し、  $P_1$  についても同様にする。このようにすれば、全体として3個のエリミネーション・フロントで計算できる。

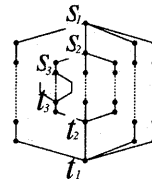


図 12. 再帰的に  $P$  グラフが含まれるグラフ

次に、  $P$  グラフの2つ以上の道に  $P$  グラフ (図 13) が含まれていたときを考える。このような  $P$  グラフを  $P$  並列と呼ぶ。ここでは  $P_1(P_2, P_3(P_4, P_5))$  を例にとって説明する。このとき  $P_2$  と  $P_3$  は並列になっている。この場合、  $P_2, P_3$  は添字の小さいほうから計算する。  $P_2$  の部分の計算をしている間、エリミネーション・フロントはこの  $P_2$  の上端点  $s_2$  と下端点  $t_2$ 、操作する辺の1つの端点の3つである。  $P_2$  の部分の計算が終わると、エリミネーション・フロントは  $s_2$  と  $t_2$  の2つである。その後、直列の時と同様にして、  $s_1$  と  $t_1$  に到達するまで辺を操作し、エリミネーション・フロントは  $P_1$  の上端点  $s_1$  と下端点  $t_1$  になる。次に  $P_3$  を計算するのだが、  $P_4$  と  $P_5$  の部分を先に計算しなければならないので、  $P_1$  の上端点  $s_1$  と下端点  $t_1$  を残したまま、  $P_4$  の上端点  $s_4$  と下端点  $t_4$  をエリミネーション・フロントに入れ、  $P_4$  の内部を計算する。  $P_4$  が終了し、  $P_3$  の  $s_3$  と  $t_3$  まで達したとき、エリミネーション・フロントは  $s_1, t_1, s_3, t_3$  であり、次に  $P_5$  の上端点  $s_5$  と下端点  $t_5$  をエリミネーション・フロントに入れ、  $P_5$  の内部を計算する。  $P_5$  の内部の計算において、エリミネーション・フロントは7点となっている。この計算を終えた後に、  $P_3$  の  $P_4, P_5$  を含む道以外を計算する。このときのエリミネーション・フロントは5点である。  $P_3$  を計算を終えたとき、エリミネーション・フロントとして  $s_3, t_3$  を持つ。  $s_3$  から  $s_1$ ,

$t_3$  から  $t_1$  についても同様に計算する。  $P_1$  についても、  $P_2, P_3$  を含む道以外で  $P_3$  と同様に計算する。このときのエリミネーション・フロントは3点である。このように  $P$  並列グラフを内部に含むとその分、エリミネーション・フロントは2つずつ増えることになる。もし、  $P_3$  の内部に  $P_4$  しかない、つまり並列でなければ、エリミネーション・フロントが増えないことに注意する必要がある。

$P$  並列の内部に  $P$  並列が何重に入っているかを並列数  $Q_0$  と呼ぶと、  $P$  グラフに対する計算の途中で現れるエリミネーション・フロントの数は  $2Q_0 + 3$  となる。以上より、  $P$  グラフに対応するリンクは並列数が少ないときは高速に Jones 多項式の計算をおこなうことができる。

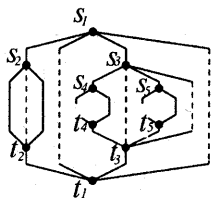


図 13.  $P$  並列を含む  $P$  グラフ

## 7 計算機実験

プレッツェルリンクについて計算機実験を行った (表 1)。プレッツェルリンクのとき、そのマイナーの数は  $B_3 = 5$  以下であるが、計算機実験では最大で 4 であることがわかった。また、アルゴリズムからその多項式の和算を行う回数  $SUM$  は  $Max\_SUM = 6 \times (\text{マイナーの数}) \times n^2$  回以下で、積を行う回数  $MULTI$  は  $Max\_MULTI = 12 \times (\text{マイナーの数}) \times n^2$  以下であることがわかる。

$SUM, MULTI$  の値からも計算時間が  $O(n^2)$  であることが確認できる。実際の計算回数が非常に少ないのは、1 つのマイナーの多項式項数を  $MAX$  では  $6n$  項としているが、実際には  $6n$  よりも少ないからである。詳細は省くが、  $P$  グラフに対応するリンクに対しても計算機実験をおこない、並列数  $Q_0$  が小さいときには有効であった。

## 8 結論

プレッツェルリンクに対する、Jones 多項式であれば記憶容量は  $O(n)$ 、計算時間は  $O(n^2)$  で求められることを示した。また、プレッツェルリンクを拡張したリンクのクラスの計算方法とその計算量を示し、  $P$  グラフとして表現できる再帰的な並列を持たないリンクであれば、  $O(n^2)$  時間で計算できることがわかった。

## 参考文献

- [1] 原正雄, 谷聖一, 山本慎: Arborescent link の Jones 多項式の計算アルゴリズムについて, 2001 年度冬の LA シンポジウム, 2002.
- [2] J. Hass, J. C. Lagarias and N. Pippenger: The Computational Complexity of Knot and Link Problems. *Proceeding of the 38th IEEE Annual Symposium on Foundations of Computer Science*, 1997, pp172-182.
- [3] J. A. Makowsky: Colored Tutte polynomials and Kauffman brackets for graphs of bounded tree width. Preprint.
- [4] J. Mignon: Knot Theory on Bipartite Graphs, *PhD thesis, Department of Mathematics, University of Toronto, Toronto, Canada*, 1999.
- [5] 関根京子, 今井浩, 今井桂子: Jones 多項式の計算, 日本応用数理学会論文誌, Vol.8, No.3, (1998), pp.341-354.
- [6] M.B. Thisletwaite: A Spanning Tree Expansion of the Jones Polynomial, *Topology* Vol.26(1987), 297-309.
- [7] D. J. A. Welsh: Complexity: Knots, Colourings and Counting, *London Mathematical Society Lecture Note Series 186*, Cambridge University Press, 1993.

表 1. プレッツェルリンクに対する計算機実験

辺の数	マイナーの数	$SUM$	$Max\_SUM$	$MULTI$	$Max\_MULTI$
5000	4	$\approx 2.05 \times 10^7$	$1.50 \times 10^8$	$\approx 5.64 \times 10^7$	$3.00 \times 10^8$
6000	4	$\approx 2.95 \times 10^7$	$2.16 \times 10^8$	$\approx 8.13 \times 10^7$	$4.32 \times 10^8$
7000	4	$\approx 4.01 \times 10^7$	$2.94 \times 10^8$	$\approx 1.11 \times 10^8$	$5.88 \times 10^8$
8000	4	$\approx 5.24 \times 10^7$	$3.84 \times 10^8$	$\approx 1.45 \times 10^8$	$7.88 \times 10^8$
9000	4	$\approx 6.65 \times 10^7$	$4.86 \times 10^8$	$\approx 1.83 \times 10^8$	$9.72 \times 10^8$
10000	4	$\approx 8.20 \times 10^7$	$6.00 \times 10^8$	$\approx 2.26 \times 10^8$	$1.20 \times 10^9$