結び目クロスキャップ数の自動計算について
On automatic computation of crosscap number of knots

山田海音, 伊藤昇 (主査), 岡本修 (副査)

Abstract

The calculation of the number of crosscap number, which has been an unsolved problem in knot theory for many years, was improved from a state where only exponential time order calculations existed to a polynomial order calculation programmatically, using a new data structure for representing knots and the S^{+}algorithm for that structure. I also implemented those algorithms in Python.

1　はじめに

数学の結び目理論では、一本のひもを捻ったり、ある部分を切断してから繋ぎなおすなどの操作をすることで、結び目の複雑さが増加していく。特に後者のタイプの操作の回数は、複雑な結び目を最初の輪の状態 (これを自明な結び目と呼ぶ。) に戻すための逆操作の回数でもある。これは工学的にも応用の効く性質である。例えばある分子の構造的な強さというのは、その分子を結び目としてみたときの複雑さに対応しているといえる。言い換えると、結び目の複雑さについて網羅した表を作ることは、結び目で表現できる現実世界のものの複雑さを理解する助けになるということである。

ある結び目について、切断してつなぎなおす操作 S^{+} が何回行われているかを、その結び目のクロスキャップ数と呼び、ある結び目 K のクロスキャップ数を n とした時、$C(K) = n$ と書くこととする。クロスキャップ数は結び目の複雑性を考える上で結び目種数と並んで"一番最初に考える数"といえるほど重要であり、クロスキャップ数の計算の高速化や自動化が求められている。クロスキャップ数 1 の結び目が求められたのは 1978 年であるが、クロスキャップ数の一般公式やアルゴリズムは未だない。困難な理由の一つは向きつき曲面に限定した種数 (穴の数) に比べ手法が見つかりにくいからである。本研究では、グラフ理論を用いて結び目のクロスキャップ数を計算する手法について提案する。

2　先行研究

2.1　$C(K) = 3$ のリストの作成

指導教員の伊藤昇先生による論文 [1] において、$C(K) = 3$ の結び目のリストが完成している。この研究では手計算で網羅的なリストを作成している。$C(K) = 2$ の結び目は 3 つ存在する。$C(K) = m + 1$ を満たす交代結び目リストを作成するには、C(K)=m のリストについて O(EC2) の計算が必要となり、m が多くなれば手計算は非現実である。クロスキャップ数に近い数であるツイスト数は交代結び目においてはジョーンズ多項式の第二係数である。しかしながらジョーンズ多項式は交点数 N とした時に $O(2^{N})$ の計算を必要とする。対象数が増える $C(K) = 4$ 以降のリストを求めるためには、クロスキャップ数計算の自動化と高速化が求められている。

2.2　KEG の考案と同値判定

先述したクロスキャップ数の計算の自動化を目的とし、伊藤先生と筆者の共著で、結び目をグラフとして再構築する論文 [2] を発表した。この研究では結び目がオイラーグラフとして解釈できることを応用し、結び目オイラーグラフ (KEG) のデータ構造ならびに、多項式時間で同値判定を行うアルゴリズムを提案した。これはS+が村杉和の一部、もしくは変形版と理解され、数学として由緒正しい操作であることを明らかにした定理を主結果としていた。一方で本研究は計算自動化を主眼とし必要なアルゴリズムを明示するものである。

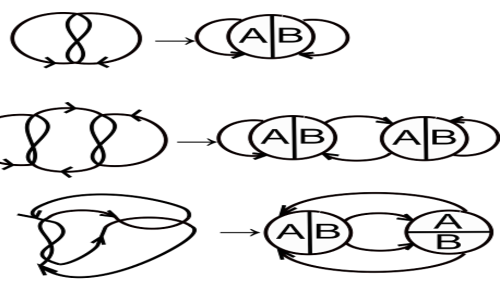

図 1: 結び目と KEG の対応

3 KEG に対する S^+ のアルゴリズム

本研究では S^+ を KEG のうちの 2 辺を選んで適用するアルゴリズムとして定義した。 S^+ の適用対象となる 2 辺の列挙と、それに先んじたグラフ自体の拡張を定義した。

3.1 グラフの拡張

ある 1 辺に対して S^+ を行いたいとき、グラフとして同一性を保ったままその辺を分割する必要がある。また、ある頂点内に対して S^+ を行いたいとき、その頂点自体を分割して辺を用意する必要がある。これは直感的には結び目を広げてほぐす操作に対応する。

3.2 S^+ の対象の列挙

拡張されたグラフについて、そのグラフに存在する辺の内 2 辺を選ぶ時のありうるペアを全て列挙する。これにより S^+ のアルゴリズムが担当すべき範囲を狭め、コード全体のメンテナンス性を向上させた。

3.3 S^+ のアルゴリズム

KEG を対象とした S^+ のアルゴリズムは以下に示す通りである。

1. Odd 頂点 O を追加する。
2. a, b から O に、O から c, d に繋がるように辺を追加する。この時 a と d, b と c がそれぞれ O の同じ側 (A/B) に繋がるようにする。即ち、(a, O, P_a, A), (b, O, P_b, B), (O, c, B, P_c), (O, d, A, P_d) の 4 辺を追加する。
3. C_e で e_s と e_g の間にある辺全てを逆向きにした辺を追加する。即ち、(u, v, P_u, P_v) を (v, u, P_v, P_u) にする。
4. $[e_s, e_g]$ の辺を削除する

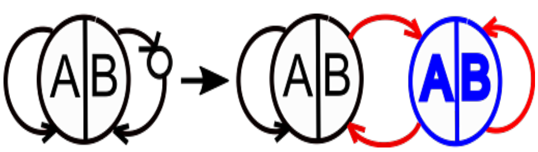

図 2: KEG に対する S^+ の適用例

4 計算量の評価

ある KEG において、任意の 2 辺に対する S^+ の結果を得るのに必要な計算量を考えていく。グラフの辺の数を E、選んだ 2 辺のうち、片方を e_s, もう片方を e_g として、

1. 頂点追加 $\rightarrow O(1)$
2. Odd 頂点への 4 辺の追加 $\rightarrow$ 定数なので $O(1)$
3. e_s と e_g の間の辺を逆にする $\rightarrow$ 最大 E-2 辺に行われるので $O(E)$
4. 逆転前の辺を削除する $\rightarrow$ 最大 E-2 辺に行われるので $O(E)$

よって、S^+ 自体は $O(E)$ で完了する。

E 辺あるうちから 2 辺選ぶ操作は、$E \times (E-1)$ ペア作れるので、全体として $O(E^3)$ で列挙が完了する。

よって、ある KEG について S^+ を行いうる部分についてもれなく S^+ を行うのにかかる計算量は、$O(E^3)$ である。

5 総括

結び目理論において長年未解決問題であったクロスキャップ数の計算について、結び目を表現する新しいデータ構造と、それに対する S^+ のアルゴリズムを用いて、指数時間オーダーの計算しか存在しなかった状態から、プログラムで多項式オーダーで計算できるよう改善した。

複雑な結び目についてもリストが作成できるようになるため、高分子化学や暗号、量子計算など、結び目を応用できる各分野について大きな貢献が見込める。

参考文献

[1] Noboru Ito and Yusuke Takimura. Crosscap number three alternating knots. *J. Knot Theory Ramifications*, Vol. 31, No. 4, pp. 2250026–1–2250026–11, 6 2022.

[2] Noboru Ito and Kaito Yamada. Plumbing and computation of crosscap number. *JP Journal of Geometry and Topology*, Vol. 26, No. 2, pp. 103–115, 11 2021.