折紙の計算量的複雑さの研究

上原　隆平＊

Computational Complexity of Origami

Ryuhei Uehara*

Key words: Computational complexity, Computational origami, NP-hard, Polynomial time algorithm

1.　はじめに

理論計算機科学と呼ばれる分野では，アルゴリズムや計算量について活発な研究が行われている.「アルゴリズム」とは「計算の方法」であり，「計算量」とは「その計算に必要な資源の量」である．つまり，この分野では問題を「こうすれば効率良く解ける」とか「どうやっても効率良く解けない」とかいったテーマに日々取り組んでいる．ここでいう効率とは，計算に使われる資源で測られる．端的に言えば，計算時間は短ければ短いほどよいし，記憶領域は少なければ少ないほどよい．100万ドルの懸賞金で有名なミレニアム問題[1]の一つである「P≠NP予想」も計算量にまつわる問題である．

「計算の方法？計算の資源？そんなの昨今のハードウェアの発展から見ると誤差範囲じゃないの？」と思う人もいるかもしれない．しかし，その認識は間違っている．幅広い最適化問題が現実的な時間で解けることで有名なソフトウェアにCPLEXがある[2]. CPLEXは，最適化問題を一般的な枠組みで記述することができれば，かなりの大きさまでそれを現実的な時間で解いてくれる．このCPLEXは，過去15年間で200万倍高速化された．そのうちハードウェアの発展による速度向上は1000倍で，アルゴリズムの改善による速度向上が2000倍であるという[2)]．つまり「計算の方法」は理論的なテーマではあるが，工学や実際の産業への応用にも大きな影響力がある．

「折紙」でいえば，「計算の効率」とは「折りの効率」と考えることができるだろう．ただ単に「折紙」といってしまうと，普通の人のイメージではこうした「折りの手間」が問題になるとは思えないかもしれない．しかし「折紙」を工学や産業へ応用する上では，今後，こうした「コスト」は重要なテーマになっていくと筆者は考えている．

例えば長い棒を折り曲げるプロセスを考えてみよう．これは3次元空間中の1次元の折り紙と見なすことができる．航空機の中の水道管など，とても長いパイプを目的の形に折り曲げる際,折り曲げる順序によっては，とてつもなく広い場所が必要になってしまう．この作業に必要な場所の広さを最小化する問題は，典型的なアルゴリズムの問題と言えよう．

もっとわかりやすい例でいえば,昨今の「コンプレックス折紙」と呼ばれる分野では，折りの手数が数百ステップに及ぶこともまれではない．こうしたステップ数を減らすことを考えるのも，それほど無駄ではないであろう．

2.　折紙の計算量的複雑さの現状

コンピュータにおける計算の効率は，計算に必要な「ステップ数」と「記憶領域」で測られる．理論計算機科学の分野では，コンピュータの抽象化を突き詰めていくと，チューリング・マシンと呼ばれるマシンモデルに行きつくが，これは現在使われているコンピュータと「本質的」に同じものと考えられている．(計算量の分野では「定数倍」，または場合によっては「多項式

＊　北陸先端科学技術大学院大学 情報科学研究科
School of Information Science, Japan Advanced Institute of Science and Technology (JAIST)

1　http://www.claymath.org/millennium/
2　http://www.cplex.com

倍」くらいの違いはほとんど気にしない．つまりスーパーコンピュータだろうがパソコンだろうが，こうしたコンピュータはみんなまとめて「フォン・ノイマン型」と呼ばれて，計算能力という点でいえばチューリング・マシンと同じと見なす．ちなみに昨今はやりの量子コンピュータは，ちょっと原理が違うので，この枠組みには乗らない．）チューリング・マシンでは非常に基本的な演算しか実行できないが，こうした基本演算を組み合わせれば，原理的にはどんなコンピュータの動作も模倣できることがわかっている．では折紙の「基本演算」と，それに基づいた「折りの効率」はどのように測られるのであろう．これについて，現時点では誰もが合意する尺度があるわけではない．しかしある程度妥当な枠組みは徐々に与えられつつある．本章ではこれらを総花的に並べて現状を紹介する．

2.1　折紙の基本操作

折紙の基本操作としては，「藤田の公理」と呼ばれる6種類の折り操作と，それに追加して「羽鳥の操作」とが標準的なものとして広く受け入れられている．**図1**の左の6個が藤田の公理であり，右の1つが羽鳥の操作である．しかし本稿では個々の操作の詳細には立ち入らないことにする．詳しくは文献10)［19章］を参照されたい．

定規とコンパスを用いた作図では2次方程式までしか解くことができず，「角の3等分問題」や「立方体倍積問題」は解くことができないことが広く知られている．一方上記の折紙の基本操作を用いれば，4次方程式まで解くことができる．つまり「角の3等分問題」や「立方体倍積問題」は折紙なら解ける．実際1970年代以降，この事実は何度も再発見されている．

また，こうした7種類の折り操作は互いに独立でなく，1つあるいは2つの折り操作を基本として，残りは「基本操作を組み合わせる」「基本操作の特別な例とみなす」ことによって実現できることが示されている[3]．

折紙の「基本操作」については，上記の枠組みと，それに関する結果がかなり広く受け入れられている．ただしこれらは，あくまで局所的な操作であることに注意する．実際の折紙では，紙をすき間に差し込んだり，凸を反転して凹にしたり，立体形状のままで「折る」操作を行ったり，複数の折りを同時に行わないといけなかったり，さまざまな状況が起こる．特に紙を曲げてよいかどうか，わずかな「のび」を利用してよいかどうか，余分な折り目を入れてよいかどうかといった点は，状況に応じて議論する必要がある．こうした状況を考慮したモデルには，いくつか異なったものがあり，モデルに応じた結果が知られている．比較的研究が行われているモデルや，妥当性のあるモデルをいくつか列挙しておく．

剛体折紙モデル： 紙の各面は剛体であると考える．折り目以外のところで折ったり，面を曲げたりすることはできない．これは，いわゆる「普通の折紙」よりも厳しいモデルである．（紙の面を一切曲げることができない折紙を想像してもらいたい．）しかし建築など，幅広い応用が見込まれるためか，最近盛んに研究されているモデルである．宇宙工学などに応用を持つ三浦公亮によるミウラ折り[4]が有名であるが，その一般化が舘知宏[5]によって精力的に行われている．

単純折りモデル： こうしたモデルの中ではもっとも単純で弱いモデル．1回の折り操作では，平坦な折り状態から平坦な折り状態へ遷移することしか許されていない．また1回の折り操作では，ある折り

3　「羽鳥の操作」の発案者である羽鳥公士郎による解説が http://origami.ousaan.com/library/constj.html にある．
4　http://www.miuraori.biz/
5　http://www.tsg.ne.jp/TT/

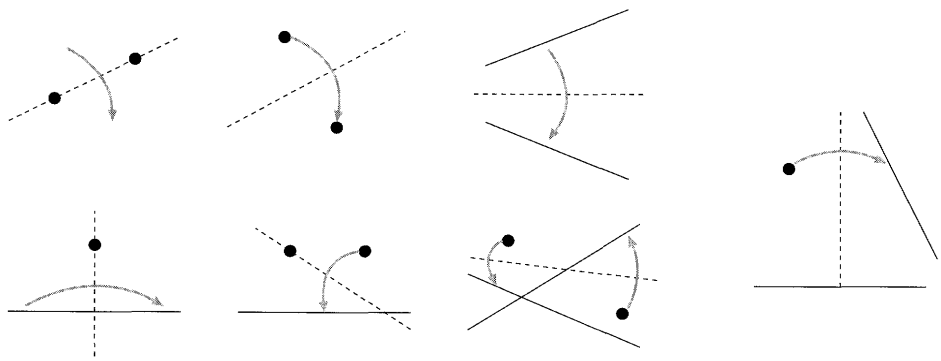

図1　藤田の公理（左の6個）と羽鳥の操作（右の1個）

目において，一番内側の紙を何枚か谷折りすることしか許されていない．(山折りは，裏返して全部を谷折りにすればよい.)このときに紙を他の紙のすき間に押し込むといった操作も許されていない．制限が強いので，アルゴリズムなどの理論的な解析が楽である．

なんでもあり： 折りに関する条件を一切つけないモデルも考えられている．このモデルでは「万能定理」が示されている．つまり，最初の状態と最後の状態が矛盾していなければ，これらの間をつなぐ「折り操作の列」が必ず存在する．これはErik D. DemaineとJoseph O'Rourkeによって示されている[10), [定理11.6.2]]が，途中で数多くの余分な折り目をつけて，紙をいくらでも小さくたためるという仮定を使っているので，理論的には興味深いが，現実的な結果とは言えない．

なお，どのモデルにおいても，途中で紙が他の紙を突き抜けないといった物理的な制約は満たさなければならない．こうした基本的な点は共通している．

2.2 折紙の時間計算量と知られている結果

現在，折紙の折りの効率に対する標準モデルはないが,「ステップ数」に対応する尺度として「折りの回数」を使うことに異論は少ないであろう．これについては非常に先駆的な研究として，田中正彦によるモデルがある[11)]．田中の提案する評価方法では，それぞれの「折り」にコストとしての数値を割り当て，それぞれの折紙モデルの折り工程の「複雑さ」を数値化して比較できるようにした．これは実際の工業製品を作る場合などの具体的な場合には役にたつ指標かもしれない．ただ，これは計算量の理論の視点から見れば「定数倍の誤差範囲」として扱われる．例えて言えばIntel社のCPUと他社のCPUの命令セットの優劣を議論するようなものであり，本稿で扱っているタイプの議論とはあまり関係ない．コンピュータサイエンスの観点からいえば，単純に「基本操作の回数」だけで測るのが妥当であろう．

2.2.1 NP困難性

計算量の理論では，ある問題が手におえないことを示す場合，NP困難性を示すことが多い．本稿ではNP困難性の定義には立ち入らないが，興味のある読者は文献5)などを参照されたい．直感的には，NP困難(あるいはNP完全)な問題は，指数関数的な組合せをすべて試さなければ，答えにたどりつけないであろうと考えられている問題である[6]．折紙のNP困難性はMarshall BernとBarry Hayesによって示された結果が代表的である[1)]．具体的には，彼らによって次の問題がNP困難であることが示された：

[入力] 正方形の紙の上に与えられたn本の山折り線と谷折り線

[質問] この紙を折り線にしたがって折り，平坦に折りたためるかどうかを判定せよ．

この問題は，(P≠NP予想が正しいとすると)本質的には可能なすべての折りかたをチェックしなければならない．ところがこのとき，紙の積み重ねの順序に指数関数的な組合せがあるため，すべてを試したのでは，現実的な時間では到底計算が終わらない．

この問題のポイントは，折る順序や紙の重なりの順序の組合せが指数関数的に爆発するところである．証明の中では，命題論理式の充足可能性判定問題を折紙で実現している．与えられた論理式を折紙の折り線で表現して，もとの論理式の条件を満たす正しい答えがわからないと，折紙が平坦に折れないようになっている．間違った答えをもとに紙を折り進めると，紙どうしの重なりに不都合が生じて，どこかで紙どうしがぶつかって平坦に折れなくなってしまう．この構成で鍵となっているのは**図2**に示した通称「ねじり折り」と呼ばれる折りかたである．図中の［>30°］と書かれたところが30°以下のときは，中心の正三角形をねじるようにして対称的に平坦に折ることができる．ところが30°よりも大きいと，こうした対称的な折りかたはできなくなってしまって，3つの放射状の帯を非対称にねじらないと平坦に折ることができない．これは普段の折紙では馴染みのない折りかたなので，頭で考えるだけではなく，是非実際に試してもらいたい．

こうした「一般には困難で手におえない問題」であっても，具体的な小さな例で，ある程度は解ける場合もある．(前出のCPLEXも，こうした「一般には難しい

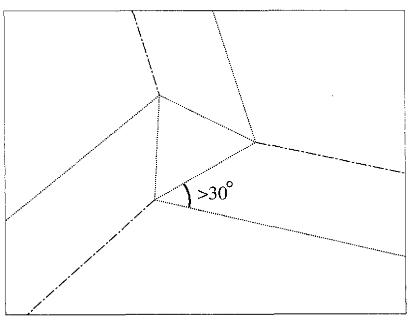

図2 三角形のねじり折り(図中の角度が30°を超える場合，山折りと谷折りを適当に非対称に変更しないと平坦に折ることはできない.)

6　P≠NP予想が成立するとすれば．

問題」をさまざまな工夫をすることで解くソフトウェアである.）折紙で言えば例えば三谷純が開発している展開図専用エディタ（ORIPA）の折りたたみ形状の予測機能は，ある程度成功している例と言える[9)].

2.2.2 多項式時間アルゴリズム

折紙の問題に対する多項式時間アルゴリズムは，まだそれほど多くが知られているわけではない．その中では，折紙の設計をサポートするフリーのソフトウェアTreeMaker[7]に実装されているアルゴリズムは抜きん出ている．このソフトウェアはRobert J. Langが10年の歳月をかけて開発してきたもので，日本の折紙業界の技術陣とも言える前川淳・川畑文昭・目黒俊幸の各氏の成果を取り込んで発展させたものである．TreeMakerでは，作りたい折紙の骨格が木の形をしていれば，それを効率よく実現する折りの展開図を自動的に生成してくれる．これを実装して実用的な速度で動作させるには，数多くのアルゴリズム上の工夫が必要であり，本稿の範囲をはるかに越える．詳しくは文献4)や10)［16章］などを参照されたい．

ここではもう少し単純な，筆者らによる「じゃばら折り」の研究を紹介する．「じゃばら折り」とはいわずとしれた「等間隔に山折りと谷折りを繰り返した折り」である．この問題は次のように定式化される：

［入力］ 長さ$n+1$の紙

［出力］ 等間隔にn箇所に山折りと谷折りを交互につける

これは一見「n回折らないとダメなのでは？」という気がするかもしれない．では次のアルゴリズムを考えてみよう．簡単のために$n=2^k-1$とする．

ステップ1： 中心で折り目にしたがって半分に折る

ステップ2： たたまれた紙の中央で重なった折り目を見て，多数決を取り，多い方にしたがって全体を半分に折る

ステップ3： 全体の長さが1になるまでステップ2を実行する

ステップ4： 全体を広げて，折り目が逆についている部分を一つずつ修正する

つまり重ねて折ることで，折る回数を節約しているわけである．このアルゴリズムを見ると，次のことがわかる．

- ステップ2と3のループは合計$k-1$回，つまりほぼ$\log n$回実行される．
- ステップ2では多数決をとっているので，毎回，少なくとも半分は正しい折り目がついている．したがってステップ4で折る回数は$n/2$以下である．

よってこのアルゴリズムの折る回数は，ほぼ$n/2+\log n$であることがわかる．$\log n$はとても小さいので，素朴な方法と比較して，だいたい2倍に改善されたことがわかる．さらに，このアルゴリズムは「じゃばら折り」だけでなく，どんな（等間隔の）折り目のパターンでも対応できる．つまり，長さnの等間隔の折り目パターンであれば，どんなものでもだいたい$n/2$回折れば，折り目をつけることができるのである．じゃばら折りに関していえば，実はもっとずっと効率のよい方法がある．具体的には，たった$(\log n)^2$回しか折らないアルゴリズムが存在する．さらにこれは，理論的にはほぼ最適で，これ以上改善できないことも示すことができる．詳細は文献3）を参照してもらいたい．

この「じゃばら折り」問題とその一般化は，もともと筆者が考案したオリジナル問題である．計算幾何学の国際会議CCCG 2008[8]の“Open Problem Session”[9]で紹介したところ好評を博し，一連の結果を得た．その中でErik D. Demaineがこの基準を“Folding Complexity”と名付けてくれた．こうした「折りの回数」すなわち“Folding Complexity”は今後，定着していくかもしれない．逆に言えば，これほど基本的な枠組みすら，これまできちんと整備されていなかったのである．

2.3 折紙の領域計算量？

折紙の領域計算量に対する標準的な基準は，筆者の知る範囲では，まだ存在しない．例えば（1次元の折紙と考えられる）長いパイプの折り曲げなどであれば「折り上げるのに必要な部屋の広さ」つまり必要な体積が問題になりそうだ．あるいは広大なシートを折る場合であれば，シートを移動する距離の総和や，そのためのエネルギーなども妥当な評価基準になるかもしれない．これらを最小化する問題は，個々の状況によっては現実的で切実な問題であろう．しかし抽象的なものも含めた「一般の折紙」を考えた場合，誰もが納得する評価基準を作るのは簡単ではない．いわゆる普通の折紙では，最初の正方形からほぼ単調に小さくなると考えられるので，そこをきちんと評価する枠組みが必要であろう．

筆者が考える妥当性のありそうな基準は，素材とし

7 http://www.langorigami.com/science/treemaker/

8 Canadian Conference on Computational Geometry 2008. http://www.cs.mcgill.ca/~cccg2008/cfp.html

9 このセッションでは未解決問題を発表・議論する．「解けていない問題を他人に発表するなんて」と思う人もいるかもしれないが，この分野は非常にオープンなのである．

ての紙の物理的な制約からくるものである．理論的な問題を議論するとき，紙の厚みは0で，いくらでも重ねることができると仮定する．しかし実際の折紙ではそうはいかない．ある折り目にたくさんの紙が挟まると，誤差が出るし，そもそも折ることができなくなってくる．破れることもある．昨今のコンプレックス折紙では，これは無視できない切実な問題になっている．こうした「紙の厚み」を考慮に入れたモデルや研究もいくつかある（例えば文献8）や http://mathworld.wolfram.com/Folding.html）が，まだまだ散発的であり，きちんとした枠組みが整備されるにはいたっていないのが現状である．

現在筆者は，こうした「折り目に挟まる紙」を最小化する問題を考えており，ある程度の枠組みと結果を出しつつある[6,7]．これについて簡単に紹介しよう．これは先のじゃばら折りの一般化問題を踏まえたもので，次の問題である．

［入力］ 長さ $n+1$ の紙と，個々の折り目における山折りと谷折りの指定

［出力］ 指定にしたがって長さ1に折りたたんだ状態．ただし折り目に挟まる紙の枚数を最小にする．

ここで「折り目に挟まる紙の枚数」について2種類の最小化が考えられる．具体的には，「最大値を最小化する」という問題と「平均値を最小化する」という問題である（なお総和を最小化する問題は，平均値を最小化する問題と本質的に同じである）．例えば山折りと谷折りの列として，［山山谷山山谷山谷谷谷谷］を考えよう．これにしたがって紙を長さ1に折りたたむ方法は，驚いたことに100通りも存在する．しかも平均値を最小にする折りかたは一通りしかなく，最大値を最小にする折りかたも一通りしかない．さらにこれらの折りかたは，互いに異なっているのだ[10]．この程度の大きさであれば，コンピュータを使った全探索で，力ずくで解くことができる．しかしもっとずっと長くなると，力ずくで解くことは難しい．筆者はこの問題はNP完全であろうと予想しているが，証明には至っていない．現在わかっているのは，以下の2点である．

- 長さ n の山／谷列をランダムに与えたとき，長さ1に折りたたむ方法は平均何通りあるか？この関数については，指数関数的であることがわかっている．求める関数を $f(n)$ とすると，実験的には $f(n) \sim 1.65^n$ であり，理論的には $1.53^n < f(n) < 2^n$ であることがわかっている[6]．
- 上記の通り，数多くの折りたたみ状態が存在するが，これらの任意の折り状態は単純折りモデルでたかだか $2n$ 回の操作で折ることができる．つまりこの問題においては，単純折りモデルは万能性をもつ[7]．

この問題は非常に基本的な問題であると考えられるが，こうした基本的な問題の困難性さえ未解決なのが，折紙の計算量的複雑さの研究の現状である．

3. 最後に

筆者は理論計算機科学，特にアルゴリズムや計算量を専門としている．こうした観点から見た折紙サイエンスの現状を紹介した．折紙の問題を考えると，計算機モデルでの「計算に必要なステップ」に対して「折りの回数＝“folding complexity”」のアナロジーはうまくいった．しかし計算機モデルでの「計算に必要な記憶領域」に対する折紙のアナロジーは，あまりうまくいっていない．しかしそれはそれで構わないと筆者は考えている．そもそもコンピュータサイエンスの成果や方法論を使いたいからといって，コンピュータの計算モデルのアナロジーにそれほど拘る必要はない．本文で紹介した「作業領域の最小化」「移動距離の最小化」「挟まる紙の枚数の最小化」は，それぞれが意味のある折紙独自の問題であり，こうした「最適化問題」は計算量の理論ではお馴染みの問題である．枠組みのアナロジーが成立しなくても，方法論が持ち込めないわけではない．

また，現在筆者らがアルゴリズムや計算量の分野で扱っているモデルと，実際の折紙の問題との間には，まだまだ大きなギャップがあることにも気づかれたであろう．「等間隔に平行な折り目がつけられた長い帯状の紙の折りかた」から，「正方形の折紙における効率のよい折りかた」へと至る道には，まだまだ数多くの問題が待ち構えている．この研究領域は，まだまだ未開拓分野であり，今後の発展が楽しみである．

10　ちなみに前者は平均1の折りかた，後者は最大3の折りかたがそれぞれ存在する．読者のみなさんは見つけられるだろうか？

参考文献

1）M. Bern and B. Hayes : The Complexity of Flat Origami, *Proc. 7th Ann. ACM-SIAM Symp. on Discrete Algorithms*, 175/183, ACM（1996）

2）R. E. Bixby : Solving Real-World Linear Programs: A Decade and More of Progress, *Operations Research*, **50**-1, 3/15（2002）

3）J. Cardinal, E. D. Demaine, M. L. Demaine, S. Imahori, S. Langerman, and R. Uehara : Algorithmic Folding Complexity, *Graphs and Combinatorics*, accepted, 2010.

（以下の国際会議版が入手可能：In *20th International Symposium on Algorithms and Computation (ISAAC 2009)*, 452/461, Lecture Notes in Computer Science, **5856**, Springer-Verlag (2009)）
4）R. J. Lang : *Origami Design Secrets*, A K Peters LTD (2003)
5）M. Sipser : *Introduction to the Theory of Computation*, PWS (1997)（邦訳：『計算理論の基礎』，太田和夫，田中圭介，阿部正幸，植田広樹，藤岡淳，渡辺治訳，共立出版 (2008)）
6）R. Uehara : Stretch Minimization Problem of a Strip Paper, *5th International Conference on Origami in Science, Mathematics and Education (5OSME)*, (2010)
7）R. Uehara : On Stretch Minimization Problem on Unit Strip Paper, *22nd Canadian Conference on Computational Geometry (CCCG)*, (2010)
8）横山卓弘，高井昌彰：厚みを持った折り紙シミュレーションとその評価，情報処理学会研究報告—グラフィックスとCAD(CG)，**2000**-115，19/24 (2000)
9）三谷純：折紙の展開図専用エディタ(ORIPA)の開発及び展開図からの折りたたみ形状の推定，情報処理学会論文誌，**48**-9，3309/3317 (2007)
10）E. D. Demaine and J. O'Rourke : *Geometric Folding Algorithms -Linkages, Origami, Polyhedra*, Cambridge (2007)（邦訳：『幾何的な折りアルゴリズム—リンケージ・折紙・多面体—』，上原隆平訳，近代科学社 (2009)）
11）田中正彦：折り紙の複雑さの数値化，折り紙探偵団，61号，11/13 (2000)