◆解説◆

「おねえさんの問題」の最先端 ―YouTube動画と世界記録―

湊　真一（北海道大学大学院情報科学研究科／JST ERATO 湊離散構造処理系プロジェクト）

「おねえさんの問題」とは

2012年8月1日より2013年4月15日までの約8カ月間，東京・お台場の日本科学未来館において，「フカシギの数え方」と題した展示[1)]が開催され好評を博した．これは筆者が研究総括を務めるJST ERATO湊離散構造処理系プロジェクトの成果展示の一環として，組合せ爆発のすごさとアルゴリズム技術の重要性を小中高生や一般市民に分かりやすく伝えることを目的としたものである．この展示物の1つとして，「おねえさんといっしょ！　みんなで数えてみよう！」というタイトルのアニメーション動画が制作され展示会場で上映された（**図-1**）．この動画はYouTubeの未来館チャンネル[2)]でも公開され，その再生回数が本稿執筆時点で140万回を超え，さらにニコニコ動画にも転載されて45万回を記録するなど，サイエンス系コンテンツではきわめて異例の大ヒット作となっている．本稿の読者でまだご存知ない方は，この機会に一度ご覧いただきたい[☆1]．

この動画では，$n \times n$の格子グラフの対角2頂点を連結する（同じところを2度通らない）パス列挙の問題を題材として取り上げている．おねえさんが子供たちの前で，パスの総数を数え上げて見せるのであるが，nが増えるに従って想像を絶する勢いで計算時間が増大していき，おねえさんは大きな困難に直面する．11×11の問題になると，スーパーコンピュータを用いても290億年（現在の宇宙の推定年齢をはるかに超える時間）かかってしまうので

☆1「フカシギ」で検索すれば容易に見つかる．

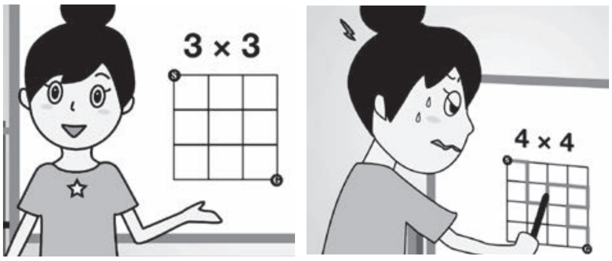

図-1　「フカシギおねえさん」の動画[2)]のカット

あるが，最先端のアルゴリズム技術を使えば，同じ問題をわずか数秒で計算できてしまうことを淡々と述べて物語は終わっている（そして来場者は，次にアルゴリズム技術の解説のコーナーに進んでいくという展示構成となっていた）．

この動画のヒットにより，実際にこの問題を解いてみようとする人がネット上に多数出現し，最近では「おねえさんの問題」と言えば，この問題を指すほど有名になっている．もしも最短パスだけを列挙するのであれば単純な組合せの問題となり，パスの総数が${}_{2n}C_n$で求められることは高校レベルで多くの人が習っている話である．ところが遠回りを許すとなると途端に難しい問題となり，パスの総数を簡潔に表す公式や漸化式などはこれまでに見つかっていない．今回のように何十桁にもおよぶ巨大数を題材とした子供向けの逸話は昔からよくあったが（たとえば一休さんが将棋盤のマス目に米粒を1粒，2粒，4粒，8粒…と置いてもらう話など），いずれも公式さえ知っていれば机上で計算できるような問題ばかりであった．それに対しておねえさんの問題は，子供にも理解できるようなシンプルな形をしているにもかかわらず，基本的にしらみつぶし的に数え上げる解法しか知られていない難しい問題であって，それでもなおかつ，アルゴリズム的な工夫によって

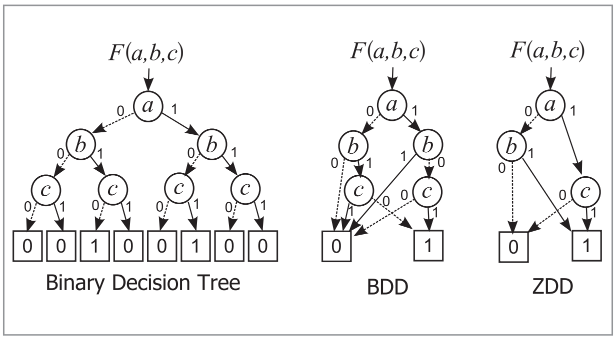

図 -2　二分決定木の例，および対応する BDD，ZDD

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

論理関数としての表現：
$F : a\bar{b}c \vee \bar{a}b\bar{c}$

組合せ集合としての表現：
$F : \{ac,\ b\}$

図 -3　論理関数と組合せ集合の対応

計算時間に圧倒的な違いが生じるという問題である．このような公式の見つかっていない問題を題材とする逸話はおそらく過去に例がなく，今回の動画が子供から専門家に至るまで幅広い人気を博した要因の 1 つと考えている．

おねえさんの問題は，本稿執筆時点で，n=25 までの答が知られている．これは筆者らの研究プロジェクトが求めた結果で，現時点での世界記録である．以下の章では，この問題を効率よく解くためのデータ構造とアルゴリズムの概要を解説するとともに，YouTube 動画の反響と今後の展望についても述べる．

データ構造「ZDD」と Knuth のアルゴリズム

おねえさんの問題の解集合を効率よく列挙するために，筆者らのプロジェクトでは，「ZDD」と呼ばれるデータ構造を用いている．以下にその概要を述べる．

二分決定グラフ（BDD : Binary Decision Diagram）[3]，1980 年代後半に論理回路などのハードウェア設計の分野で考案された論理関数データのグラフによる表現である．これは，論理関数の値をすべての変数について場合分けした結果を二分木グラフ（Binary Decision Tree）で表し，これを簡約化することにより得られる．**図 -2** に，a，b，c の 3 変数を入力とする論理関数を二分木グラフで表現した例と，それを簡約化した BDD の例を示す．図中に示す通り，二分木の根節点からスタートして，各分岐節点に書かれている変数の入力値に応じて，0 または 1 のラベルのついた枝（0- 枝または 1- 枝）のいずれか一方に進んでいくと，最終的に関数の出力値を表す終端節点（0- 終端節点または 1- 終端節点）に到達するようなグラフになっている．ここで，BDD の簡約化を行う際に，場合分けする変数の順序を固定し，(1) 冗長な節点を削除する，(2) 等価な節点を共有する，という処理を可能な限り行うことにより「既約」な形が得られ，論理関数をコンパクトかつ一意に表せることが知られている．BDD によるデータ圧縮率は，論理関数の性質にも依存するが，例題によっては数十倍～数百倍以上もの圧縮率が得られる場合がある．さらに圧縮率にほぼ比例してメモリ量だけでなく計算時間も削減できるという特徴を持つ（詳細は文献 4)，5) 等を参照）．

BDD は，元々は論理関数の表現法であるが，n 種類のアイテムから任意個選ぶ組合せの集合を表現することもできる．**図 -3** は，図 -2 で例示した論理関数を真理値表で表現したものであるが，これは ac および b という 2 つの組合せを要素に含む集合表現と見ることもできる．このような組合せ集合データの処理に特化した BDD として**ゼロサプレス型 BDD（ZDD または ZBDD : Zero-suppressedBDD）**[6] が知られている．ZDD は，通常の BDD と異なる簡約化規則を持つ．すなわち，**図 -4** 右に示すように，1- 枝が 0- 終端節点を直接指している場合に，この節点を取り除く．その代わり，通常の BDD で削除されるような節点（同図左）はあえて削除しない．先ほどの図 -2 の例では，同じ二分決定木から簡約化した ZDD も併記している．ZDD の簡約化規則は，特に疎な組合せの集合に対して顕著な効果が

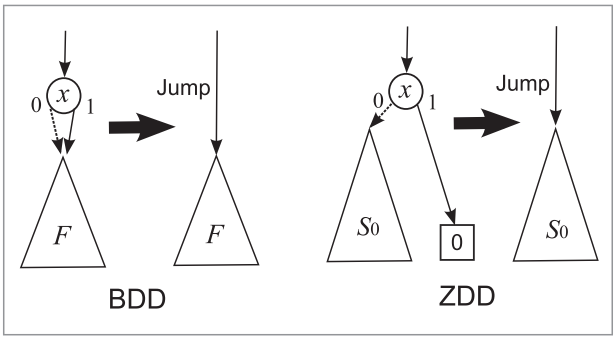

図 -4　BDD と ZDD の簡約化規則

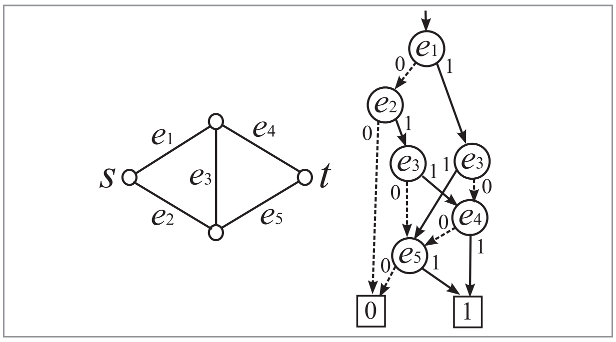

図 -5　s, t 間のパスを列挙する ZDD

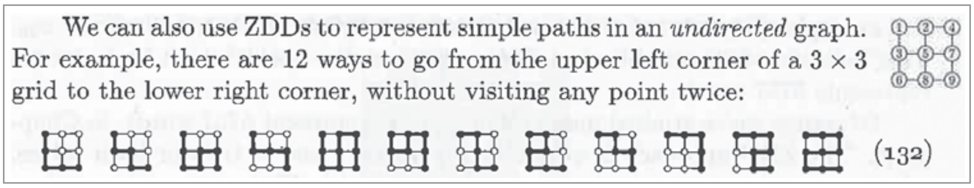
We can also use ZDDs to represent simple paths in an *undirected* graph. For example, there are 12 ways to go from the upper left corner of a 3 × 3 grid to the lower right corner, without visiting any point twice: (132)

図 -6　Knuth の教科書 [7] **に記載されている Simpath アルゴリズムの実行例．文中に 3 × 3 とあるのは頂点数で，格子数では 2 × 2 に相当する**

ある．たとえば，アイテムの平均出現頻度が 1% であれば，ZDD は BDD よりも 100 倍大きな圧縮率が得られる可能性がある．ZDD は筆者が今から約 20 年前に初めて考案し命名したデータ構造であるが，BDD の種々の派生形の中で最も重要なものとして認識されており，Knuth の有名な教科書“The Art of Computer Programming”の最新巻 [7] でも独立した項目として詳しく解説されている．

ZDD は，グラフの部分集合を列挙索引化するためのデータ構造としても有用である．節点集合 $V=\{v_1, v_2, \cdot\cdot\cdot, v_n\}$，辺集合 $E=\{e_1, e_2, \cdot\cdot\cdot, e_m\}$ からなるグラフ $G=(V, E)$ を考えると，グラフ列挙の問題とは，E のべき集合 2^E（または V のべき集合 2^V）の中から，ある条件を満たすような部分集合を求めることであり，これは解集合を辺（または節点）の組合せの集合と考えれば，そのまま ZDD で表現することができる．たとえば，**図 -5** ではグラフの 2 節点 s, t を結ぶパスの集合を表す ZDD を示している．この例では s, t 間のパスは 4 通り存在するが，それぞれのパスを辺の組合せと考えれば $\{e_1e_4, e_1e_3e_5, e_2e_5, e_2e_3e_4\}$ という組合せの集合として解集合を表現できる．これを表す ZDD では，最上位の根の節点から 1 の終端節点に至るパスが 4 通り存在する．ここで，ZDD の 0- 枝はグラフ G の当該辺を使わないことを意味し，1- 枝は当該辺を使うことを意味すると解釈すれば，ZDD の各パスがこの問題の解に対応していることが確かめられる．

先に述べた Knuth の教科書（Vol.4 Fascicle 1）[7] の p.121（Vol.4A では p.254）において，与えられたグラフの 2 点 s, t を結ぶ単純パス（遠回りを許すが同じ頂点は 2 度通らないパス）を全列挙するアルゴリズム Simpath が記述されている．これは，s-t 間の単純パスとなるような辺の組合せを全列挙した ZDD を構築するアルゴリズムであり，まさに，おねえさんの問題を解く方法である（**図 -6** に同書の紙面の一部を示す）．さらに，Knuth 本人が実装したソースコードが氏の Web ページで公開されているが，試してみるとこれが驚くほど高速である．たとえば 14 × 14 の格子グラフ（辺の総数 420）の対角 2 頂点を結ぶ単純パスを全列挙する ZDD を生成させると，パスの総数は実に 227449714676812739631826459327989863387613323440（約 2.27×10^{47}）通りもあるが，ZDD の節点数はたった 144759636 個で済んでおり，その ZDD 生成と解の数え上げに要する時間はわずか数分である．

図 -7 により Simpath アルゴリズムの大まかな処理手順を示す．まず与えられたグラフ G の辺に適当な順序をつけ，$E=\{e_1, e_2, \cdot\cdot\cdot, e_m\}$ とする．これより，根節点から順に $e_1, e_2, \cdot\cdot\cdot$ の変数順の二分決定木を，上位から下位に向かって幅優先順で場合分けしながらトップダウンに構築していく．この幅優先順の処理の各段階で，グラフ G における処

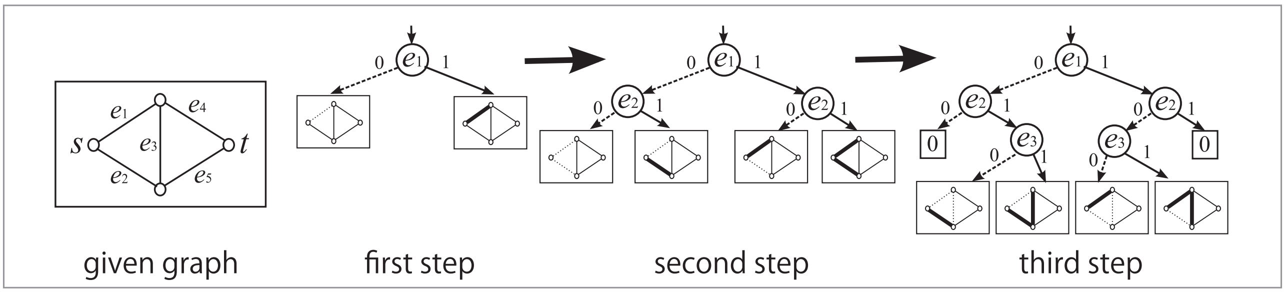

図 -7　Simpath アルゴリズムの処理手順

理済みの辺と未処理の辺の両方に接続している頂点の集合を Knuth はフロンティア（frontier）と呼び，このフロンティアを始点 s から終点 t まで徐々に移動させながら，フロンティア上の頂点の辺の接続情報を途中状態として記憶して，動的計画法により等価な状態をまとめて共有することにより，最終的に圧縮された ZDD を高速に構築している．処理途中に共有した状態数を加算していくことにより，解の数え上げも ZDD 構築と同時に完了する．計算時間は構築した ZDD の節点数にほぼ比例する（より詳細については文献 8）を参照されたい）．

Knuth はさらに，Simpath アルゴリズムの一部を変更するだけで，s-t 間の単純パスだけでなく，ハミルトンパス，有向パス，さらに各種のサイクルを列挙する ZDD もほぼ同様に構築できることを述べている．さらに途中状態の記憶の仕組みを改変することで，連結部分グラフの列挙，大域木／林の列挙，カットセットの列挙，グラフの k 分割問題等，多くの問題に適用できる．筆者らはこのような幅優先トップダウン型の動的計画法による ZDD の構築法を総称して「フロンティア法」と呼び，さまざまな実問題への応用を試みている．

一般にフロンティア法では，処理途中でフロンティアが膨らむと，互いに異なる途中状態が多く発生して計算量が増大するため，フロンティアがなるべく小さいままで処理が進むことが望ましい．フロンティアが小さくて済むかどうかは，与えられたグラフの形と，それに応じた辺の順序付けに依存する．一般には平面グラフに近くて細長い形のグラフである方が都合がよい．おねえさんの問題で扱われている $n \times n$ 格子グラフは，フロンティア法による圧縮効果が非常に大きくなるグラフの一例である．

日本科学未来館での成果展示と YouTube 動画の反響

さて，冒頭で述べた通り，上記のパス列挙の問題が，おねえさんの問題として知られるようになったのは，未来館での成果展示で制作した YouTube 動画に由来するが，そもそものきっかけは，未来館を管轄している JST 本部から我々の ERATO プロジェクトに対して出展を打診されたことであった．ただしあくまでも未来館側と ERATO 側の責任者同士を引き合わせるだけで，実際に出展できるかどうか，また受け入れ可能かどうかは現場の判断に任されていた．筆者らは未来館の担当者と何度か会合を持ち，プロジェクトの研究内容を紹介したところ，その中でも特に，「那由他」や「不可思議」の単位に到達するような超巨大数を正確に数え上げるアルゴリズム技術に非常に興味を持たれて，これなら未来館で展示できそうだということになり，実際に動き出すことになった．

近年，学術的な研究成果を社会に還元するアウトリーチ活動が重要視されており，特に ERATO のような大型プロジェクトでは積極的な広報活動が求められている．未来館は日本国内での知名度は抜群であり，ここで成果展示を行うことは，プロジェクト研究員や学生にとっても大いに士気が上がる話であった．ただし，未来館での展示は成功しても失敗しても影響が大きいので，いい加減な仕事は許されない．基本的にプロジェクトリーダが直接かかわって十分な時間をかけて準備作業にあたることにした．

未来館には，宇宙地球科学，生命科学，情報科学などの分野の展示がある．一般に自然科学系の展示は，たとえばロケットエンジンや脳の断面図のように，実物や模型を見せることによって，研究者が挑戦していることを子供にも直接的に訴えやすいのに対し，情報系の展示は，何となく未来的なものは作れても，結局何を訴えたいのかが分かりにくい傾向がある．特に，アルゴリズムのような抽象的なものを小中高生や一般市民に分かりやすく見せる展示は，これまでにほとんど例がなく，展示方法から考える必要があった．未来館は，これまでにない新しい展示手法を開発し，全国の科学館に展開していくことがミッションの1つとなっているため，展示開発課の若手スタッフの方々を中心に積極的に取り組んでいただいた．さらに未来館には一流の空間デザイナやメディアアート系クリエータの人たちが出入りしており，彼らとのコラボにより面白い展示作品が生まれることになった．

$n \times n$ 格子グラフの n を淡々と増やしていく手法は，アルゴリズムに関しては専門外であるクリエータやデザイナの方々の意見によるところが大きい．研究者の立場からは，何とかして興味を持ってもらおうとして，グラフを京都や札幌の街路に例えたり，テーマパークのアトラクションを回る問題に例えるなど，いろいろな案を検討したが，結局，「変に細工しなくてもこのままで十分面白いんじゃないですか」と言われて，そう言われればそうかということになり，格子グラフの問題をそのままシンプルに見せることにした．

この展示では，抽象的な内容を実感させるため，タブレット端末を用いたインタラクティブな展示手法や，人間の視野に入り切らないような巨大な数表を掲示して迫力を与える手法，体重をかけることによりグラフが圧縮される体感展示の手法，若手研究者たちの顔写真を掲示して研究の面白さを語らせる手法など，多彩な手法を組み合わせた展示構成となった．「フカシギおねえさん」の動画もそのような展示方法の1つとして制作された．情報科学の世界では，問題が解けるかどうかだけではなく計算時間が重要である．これを示すために巨大数をひたすら読み上げることでその処理時間を意識させることと，人間の寿命と対比して計算時間の重要性を再認識させることを意図して，動画の制作を依頼した．担当ディレクタが作成したストーリーでは，当初からおねえさんが子供に教えるというアイディアは出ていたが，スパコンで16 × 16の問題を実行したら日が暮れて子供があきれるという常識的なオチが書かれていた．その場で筆者が少し机上計算して「この想定だと10 × 10でもスパコンで数十万年かかりますね」と言ったところ，製作スタッフの人たちが「そんな例え話作るの無理ですよ」と倒れそうになっていた．その3日後にディレクタが作って持ってきたのが，おねえさんがロボットになるという壮絶なストーリーだった．提案した本人はシュール過ぎて採用されるわけがないと思っていたらしいが，関係者一同，これは面白いということで，制作に取り掛かることとなった．実写にするかアニメにするかを決める必要があったが，絵コンテのデザインが良かったのでアニメでの制作を依頼した．その後，アニメ制作スタッフと声優の皆さんががんばってくれて非常に良い作品に仕上がった．

おねえさんの動画はもちろんフィクションであるが，パスの総数と計算時間は技術的に考証したもので，スパコンで毎秒2億通りのパスを出力できるような出力線形時間のアルゴリズムを想定している．動画公開後，グラフをバックトラック探索して解くプログラムを作ってみたという一般市民プログラマからの報告が相次いだが，その方法では，おねえさんとほぼ同じ時間がかかることになる．なお，通常のバックトラック型の探索では，得られたパスを次々に表示したときに，終点付近のみが変化し始点に近い部分がほとんど動かないため，見ていて面白くない．そこで，わざとランダムな順序で解を選び出し画面に表示するという細かい演出を加えている．我々の手法では解集合がZDDの形で索引化して格納されているため，膨大な個数の解の中から一様ランダムな解を高速に選び出せる．

さて，この動画がYouTubeで公開されると，想

像を超える壮大なストーリーがネット上で話題を呼び，最初の 24 時間で 4 万ビュー，1 週間で 70 万ビューを記録した．さらにニコニコ動画にも転載されて急激に再生回数が伸び，2012 年 9 月第 3 週には全カテゴリを合わせた総合週間ランキングでも首位を獲得した☆2.

この動画の反響は日本国内にとどまらなかった．未来館の展示物には原則として英語訳をつけることになっており，おねえさんの動画にも英語字幕が入っている．このため海外からも多くのアクセスがあり，コメント欄には英語，イタリア語，ロシア語，中国語，韓国語，アラビア語など，世界各国語での書き込みが多数見られた．さらには中国の動画サイトにも勝手に転載されて，中国語の字幕や大量のコメントが付けられてこれまでに 6 万回以上再生されている．おねえさんの問題はきわめてシンプルなので，言葉が通じない海外の人でも容易に理解できるものと思われる．筆者は国際会議に参加した際にも，この動画を見せることがしばしばあるが，おおむね好評であり，帰国後に学生に見せたいので URL を教えてほしいという反応をいただくことが多い．

筆者は動画公開の際に，Knuth ご本人にも動画作成の意図についてメールで報告していた．後日，我々のプロジェクトの研究員が Knuth の教科書の訂正事項を見つけて報告し，そのお礼の小切手と手紙をいただく機会があったが，その末尾に「Best wishes to you and Shin-ichi for 2013. I enjoyed the YouTube video about big numbers, and shared it to several friends.」と書かれていた．大御所の Knuth からも「お墨付き」をいただけたということで，展示にかかわったプロジェクト研究員や未来館の制作担当者にとっては，たいへん嬉しい知らせであった．

未来館での展示「フカシギの数え方」は，会期中に 23 万人を超える来場者を集め，好評のうちに終了した．現在では，2013 年 7 月より北海道大学総合博物館に会場を移して，引き続き同タイトルでの展示を行っている．

☆2 本件は好評な話題なのでよかったが，失言や不祥事で炎上すると大変なことになるので，我々は「ネット爆発は組合せ爆発より怖いので注意すべし」と自らを戒めている．

おねえさんの問題の世界記録

我々の研究プロジェクトでは，未来館の展示と並行して，おねえさんの問題を計算できる限界の規模への挑戦を進めてきた．ERATO 研究員の岩下らは，Knuth の実装した Simpath アルゴリズムを，メモリ効率がよくなるように改善し，約 500GB のメモリ容量のマシンを用いて，18 × 18 までの ZDD の生成に成功していた．n=19 以降はメモリ容量が不足するので，幅優先順の横 1 列分だけの ZDD 節点を生成しながら，解の個数だけをカウントするプログラムを開発し，n=21 までの数え上げに成功した．この結果は 2012 年 9 月上旬にオンライン数列百科事典サイト The On-Line Encyclopedia of Integer Sequences（OEIS）[9] に申請し公式に登録された．これまでの記録は n=19 だったので一気に 2 段階更新したことになる．ちなみに n=12 の結果は 1995 年に Knuth が初めて記録しており，由緒ある問題であることが分かる．さらに面白いことに，我々がこのサイトを見に行ったときには，おねえさんの YouTube 動画が参考文献としてすでにリンクされていた．フィンランドの人が動画を見つけてリンクすることを提案して認められたらしい．

世界記録を保持するということは，実用的な貢献とは別に，技術の先進性を端的に示すことで，一般市民や研究者に夢や動機を与える効果がある．たとえば，見学に来た高校生に世界記録の話をすると目が輝き出すという反応もしばしば見られることである（もちろん，何でもただ 1 番になればよいというものではなく，1 番になった後にどう役立つのかも大事であるが）．

我々のグループでは，n=21 の世界記録達成後も，さらなる記録への挑戦を続行した．n=21 までは，どのような形のグラフでも計算できる汎用的なプログラムを用いていたが，グラフの形を $n \times n$ 格子グラフに限定すれば，それに特化したアルゴリズムを工夫することができ，計算時間やメモリ量を抑え

n	パスの総数
1	2
2	12
3	184
4	8512
5	1262816
6	575780564
7	789360053252
8	3266598486981642
9	41044208702632496804
10	1568758030464750013214100
11	182413291514248049241470885236
12	64528039343270018963357185158482118
13	69450664761521361664274701548907358996488
14	227449714676812739631826459327989863387613323440
15	2266745568862672746374567396713098934866324885408319028
16	68745445609149931587631563132489232824587945968099457285419306
17	634481461123796397131029754079552440044944398686648069364636938 7855336
18	1782112840842065129893384946652325275167838065704767655931452474605826692782532
19	152334497170487999308074281031922969089945425532329455577602986673735506059287756925 5844
20	396289219982303756020729951713336250210633970573946377151523711337701068236403570670447206494 0398
21	3137475105013710272042053813738221451310331219369872365306135199134643337938938579396557699224602131646 3868
22	755970286667345339661519123315222619353103732072409481167391410479517925792743631234987038883317634987271171404439792
23	55435429355237477009914318489061437930690379970964331332556958646484008407334885544566386924020875711242060085408513482933945720
24	12371712231207064758338744862673570832373041989012943539678727080484951695515930485641394550792153037191858028212512280926600304581386791094
25	8402974857881133471007083745436809127296054293775383549824742623937028497898215256929178577083970960121625602506027316549718402106494049978375604247408

表 -1　おねえさんの問題の世界記録 9)

られるため，より大きなグラフでも計算が可能となる．具体的には，パスが交差しないことを利用したデータ構造の簡単化や，格子グラフでは横の辺の状態だけを管理すれば十分であることを利用した簡単化，およびグラフの規則性に基づいて最小完全ハッシュ関数を構成する方法など，いくつかの改善手法を開発した．さらに中国人剰余定理を用いて超巨大数を記憶するメモリ量を節約する方法，マルチコア並列による高速化，グラフの対称性を利用した問題分割法なども適宜使用している（詳細は文献 10)，11）を参照)．

2013 年の正月には，一般市民のプログラマから「おねえさんの問題の高速な解き方を思いついたので見てほしい」とのメールがあり，たまたま札幌在住の方だったので，休日に何度か北大に来ていただいて議論したところ，中にはプロも驚くようなアイディアも含まれており，互いに大いに知的刺激を受けた．彼には「アマチュアプログラマー」の肩書で，上記文献の共著者にも加わってもらっている．これは日本における情報科学技術の裾野の広がりを示す出来事とも言える．江戸時代にも，庶民が算術の問題を解いて神社に奉納していたと聞いているが，そのような和算の伝統が現代まで続いているように思われる．

以上のようなさまざまな工夫により，計算能力はじわじわと向上し，2013 年 1 月末には n=24 までの計算が可能になっていた．技術ドキュメントの執筆が終わり次第，登録申請を行う予定であったが，2013 年 2 月に OEIS サイトを確認したところ，いつの間にか，ノルウェーの大学チームが一気に n=24 まで記録更新していたことが分かった．そこで，我々は総力を結集して n=25 までの計算に取り組み，2013 年 4 月に世界記録の奪還に成功した．これは上記のさまざまなアルゴリズム的な工夫の末に，主記憶 1.5TB のマシンを数日間占有して計算したもので，今後短期間では破られないだろうと予想している．

表 -1 に最新の計算結果を示す．n=24 までは，我々の計算結果と他の研究グループの結果が完全に一致しているので数値の正しさは確かめられているが，n=25 はまだ我々しか計算に成功していないので，この数値は現時点では登録されただけで確かめられたものではない．

計算結果の表からは，10 進数の桁数はおよそ n^2 の勢いで伸びているように見える．つまりパスの総数は，グラフの格子数 n^2 に対して指数的に増大していると思われる．これに対して，格子グラフに特化したフロンティア法を用いると，処理中のフロ

ンティアの最大状態数は $O(3^n)$ で抑えることができ，処理時間もこの状態数にほぼ比例する．フロンティア法により，指数の肩に乗る数を n^2 から n に減らせるので，おねえさんの使ったナイーブな方法に比べて圧倒的な速度改善効果が得られる．それでも n を増やせば限界に突き当たることに変わりはないが，計算可能な範囲が大きく広がっていることは確かであり，現実に解きたい重要な問題がその範囲に入ってくれれば，アルゴリズム技術の果たす意義は非常に大きいと言える．

今後の展望

文献 8)，12）でも述べている通り，フロンティア法は単に世界記録を出すためだけの技法ではなく，配電網や道路網，鉄道網，ガス，水道，通信網などの解析や制御，さらに避難所の配置や，選挙の区割り問題など，社会的に重要なさまざまな応用に深くかかわっている．一般に，電力網や交通網のような社会インフラ系の構造は，平面グラフや格子グラフに近い形であることが多いので，フロンティア法による圧縮効果がきわめて高くなる傾向がある．筆者らは，電力網制御の専門家である早稲田大学・林泰弘教授との共同研究で，現実的な配電網のモデルで変電所の給電パタン候補を列挙する問題をグラフ k 分割問題として定式化し，フロンティア法により求める手法を開発した[12]．実験の結果，今まで現実的に不可能と考えられていた 10^{71} 通りもの膨大な給電パタンを全列挙する ZDD をわずか 1 秒程度で生成できた．2011 年 3 月の大震災以降，電力網の解析と制御の重要性が増大していることから，今後も本手法に関連する研究開発を継続していく予定である．

我々の研究グループでは，データ構造 ZDD とフロンティア法の技法を用いて，ある制約を満たすグラフの部分集合を全列挙し索引化する処理系を開発し，さらにこれに Python をベースとした使いやすいインタフェースを装備したものを「Graphillion」[13] という名前で公開している．Graphillion のチュートリアル動画は，おねえさんの動画の続編風に作られており，一部の人々には好評を博している．興味のある方は一度ご覧いただければ幸いである．

参考文献

1) 日本科学未来館メディアラボ第 11 期展示「フカシギの数え方」（2012），http://miraikan.jp/medialab/11.html
2) 土居誠史 他：『フカシギの数え方』おねえさんといっしょ！みんなで数えてみよう！（2012），YouTube video, http://www.youtube.com/watch?v=Q4gTV4r0zRs, http://www.nicovideo.jp/watch/sm18847458
3) Bryant, R. E. : Graph-based Algorithms for Boolean Function Manipulation, IEEE Transactions on Computers, Vol.C-35, No.8, pp.677–691 (1986).
4) 藤田昌宏，佐藤政生：特集 BDD（二分決定グラフ），情報処理，Vol.34, No.5, pp.584–630 (May 1993).
5) 湊 真一：BDD（二分決定グラフ）とその応用，応用数理，Vol.9, No.3, pp.194–206 (1999).
6) Minato, S. : Zero-suppressed BDDs for Set Manipulation in Combinatorial Problems, In Proc. of 30th ACM/IEEE Design Automation Conference (DAC' 93), pp.272–277 (1993).
7) Knuth, D. E. : The Art of Computer Programming : Bitwise Tricks & Techniques ; Binary Decision Diagrams, Vol.4, fascicle 1. Addison-Wesley (2009).
8) 湊 真一：BDD/ZDD を用いたグラフ列挙索引化技法：特集 BDD/ZDD を用いた新しい列挙索引化技法（フロンティア法）とその応用），OR 学会誌，Vol.57, No.11, pp.597–603 (2012).
9) Number of Nonintersecting (or Self-avoiding) Rook Paths Joining Opposite Corners of an nxn Grid, the On-line Encyclopedia of Integer Sequences, https://oeis.org/A007764
10) Iwashita, H., Nakazawa, Y., Kawahara, J., Uno, T. and Minato, S. : Efficient Computation of the Number of Paths in a Grid Graph with Minimal Perfect Hash Functions, Hokkaido University, Division of Computer Science, TCS Technical Reports, Vol.TCS-TR-A-10-64 (2013).
11) 岩下洋哲，中澤吉男，川原 純，宇野毅明，湊 真一：最小完全ハッシュ関数を用いたグリッドグラフ上の効率的なパス数え上げ，情報処理学会アルゴリズム研究会，情処研報，Vol.2013-AL-143, No.8 (2013).
12) 井上 武 他：フロンティア法による電力網構成技術：特集 BDD/ZDD を用いた新しい列挙索引化技法（フロンティア法）とその応用，OR 学会誌，Vol.57, No.11, pp.610–615 (2012).
13) Inoue, T., et al. : Graphillion (2013), http://graphillion.org/

（2013 年 8 月 13 日受付）

謝辞　研究成果展示にご協力いただいた多くの関係者の方々に感謝いたします．特に，未来館展示担当の鈴木真一朗氏，相川直美氏，空間デザインの中原崇氏，展示作品制作の古堅真彦先生，映像ディレクタの土居誠史氏に感謝いたします．

湊 真一（正会員）minato@ist.hokudai.ac.jp

1988 年京大・工・情報工学科卒業，1990 年同大学院修士，1995 年博士（社会人）修了．博士（工学）．1990 年日本電信電話（株）入社．NTT 研究所にて大規模論理データ処理アルゴリズムの研究に従事．2004 年北大・情報科学研究科助教授．2010 年より同教授．2009 年より JST ERATO 湊離散構造処理系プロジェクト研究総括（兼務）．BDD（二分決定グラフ）を用いた離散構造の処理に興味を持つ．著書 “Binary Decision Diagrams and Applications for VLSI CAD”（Kluwer，1995 年）．電子情報通信学会，IEEE 各会員．