

フロンティア法：BDD/ZDDを用いた 高速なグラフ列挙索引化アルゴリズム

湊 真一

北海道大学 情報科学研究科 / JST ERATO

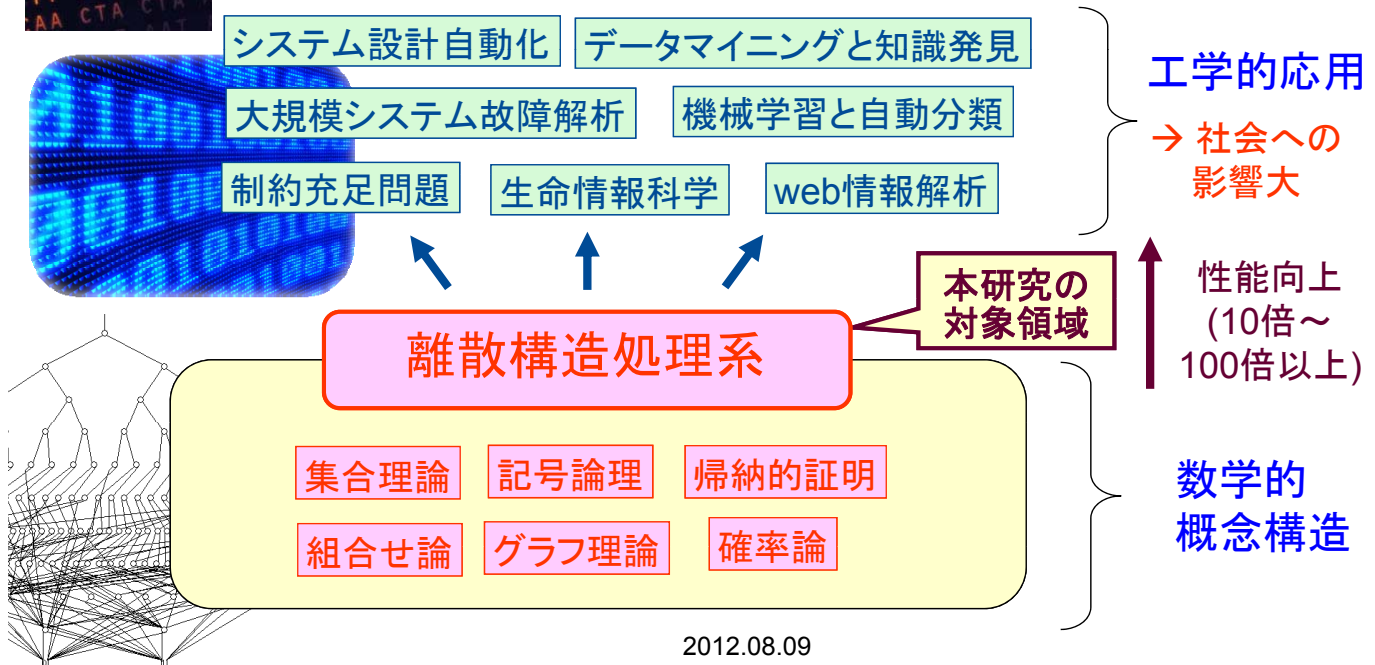
2012年8月9日

ERATOとは

- JSTの戦略的創造研究推進事業
 - さきがけ(牧場型)、CREST(ハヶ岳型)、**ERATO(富士山型)**
- ERATOプロジェクトの特徴
 - 新しい科学技術の源流を作るような研究を支援。
 - 昭和56年発足。過去に98プロジェクトを採択。
(科学技術の全分野で年4~5件)
 - プロジェクト期間:5年半 研究費総額:10~15億円(テーマによる)
 - メンバ規模:10~15人(人件費に依存)
 - 研究総括に自由裁量を与え、分野・組織にとらわれずに機動的なプロジェクトを構成。
 - 単なる助成金ではなく、JSTが主体的にプロジェクトを推進。
本務研究室とは別に独自のオフィスを設置。

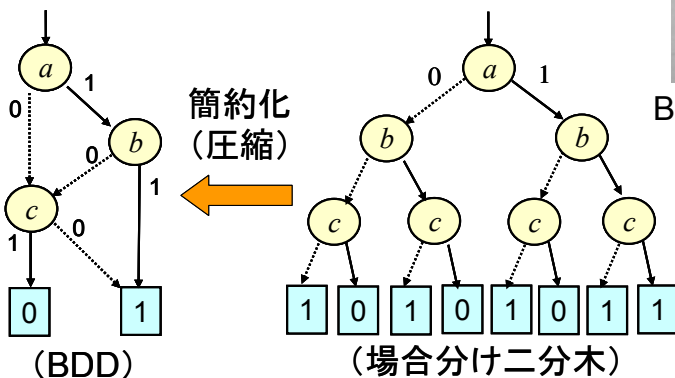
本プロジェクトの基本構想

■ 様々な工学的応用を持つ基盤技術として
「離散構造処理系」に着目し、研究開発を行う



BDD(二分決定グラフ)

離散構造の最も基本的なモデルである**「論理関数」**の処理技法

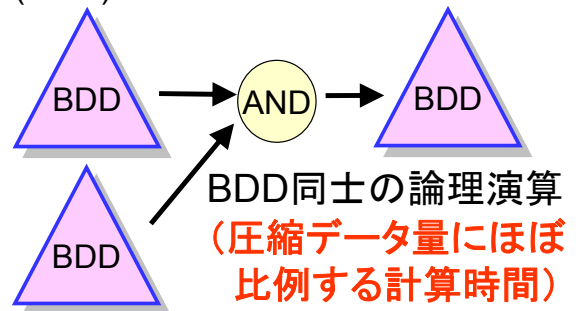


- ・ 場合分け二分木グラフを簡約化(データ圧縮)
- ・ 多くの実用的な論理データをコンパクトかつ一意に表現。(数十~数百倍以上の圧縮率が得られる例も)



Bryant (CMU)

1986年に画期的なBDD演算 (**Apply演算アルゴリズム**)を提案。以後急速にBDD技術が発達。(長期間、情報科学の全分野での最多引用文献となった)

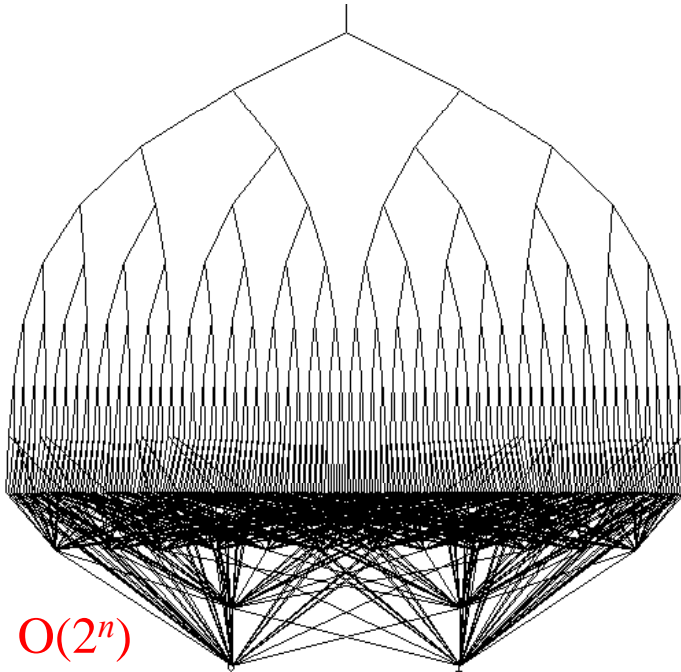


近年のPC主記憶の大規模化により、BDDの適用範囲が拡大 (特に2000年以降)

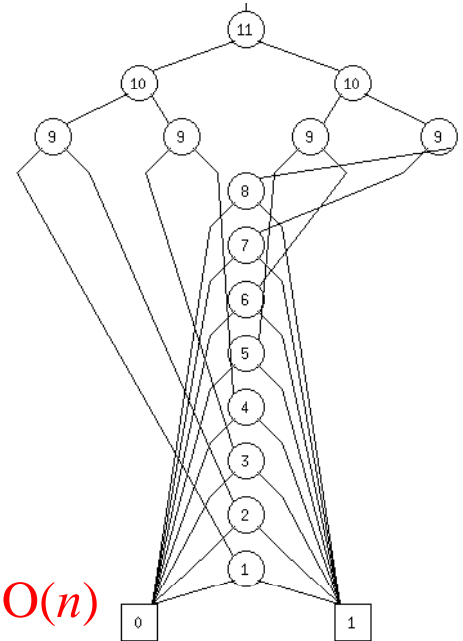


BDD簡約化の効果

- 特定の問題では、指数関数的な圧縮効果が得られる。
 - 例題に依存するが、多くの実用的な問題で効果がある。



$O(2^n)$



$O(n)$

2012.08.09

5

論理関数と組合せ集合

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

論理関数:

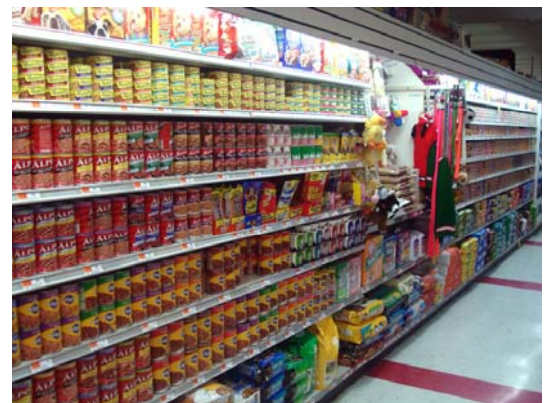
$$F = (a b \sim c) \vee (\sim b c)$$

組合せ集合:

$$F = \{ab, ac, c\}$$



(買い物客の購入品)



→ ab

→ c

→ ac

- 組合せ集合と論理関数の演算は対応関係がある。

- Union of sets → logical OR
- Intersection of sets → logical AND
- Complement set → logical NOT

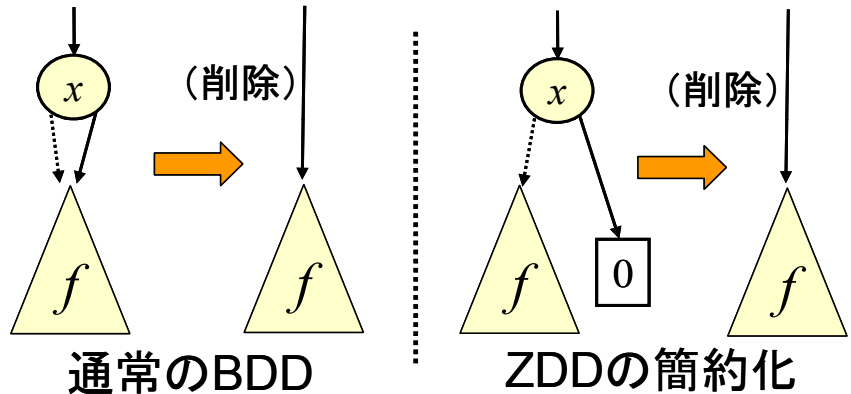


6

ZDD(ゼロサプレス型BDD)による集合の表現

■ 「組合せ集合」を効率的に表現するためのBDDの改良

- 湊が世界で初めて考案し命名(1993年)
- 通常と異なる簡約化規則を考案。
- 疎な集合の族を扱う場合に著しい効果が得られる。
(例: 商店の陳列アイテム数に比べて1顧客の購入点数は極めて少ない。)



■ ZDDはBDDの改良技術として現在、世界的に広く使われている。

- 最近では、データマイニング分野に応用されて、画期的な有効性が示されている。(数百倍のデータ圧縮率・数十倍の処理高速化)
- 他にも応用例は増えつつある。



2012.08.09

7

ZDDの応用

- 元々はLSI設計の論理式(CNF/DNF)の単純化に利用
 - 膨大な項数の論理式を高速に因数分解する方法[Minato96]
 - 算術式の因数分解法[Minato97]
- データマイニングへの応用
 - ZDDを用いた頻出パターンマイニング[Minato2005]
 - LCM over ZDDs アルゴリズム[Minato-Uno2008]
 - 時分割データベースからのパターン変化の検出[Minato2010]
- グラフに関する種々の問題への応用
 - 最大クリーク問題、彩色問題、カバー問題等[Coudert97]
 - ZDDを用いた超高速パス列挙アルゴリズム[Knuth2009]
 - 電力ネットワークの制御、避難所配置問題、通信NW、他



2012.08.09

8

ZDD応用例: 頻出アイテム集合抽出問題

■ データマイニングの最も基本的な問題

- 最小出現頻度 α 以上のレコードに含まれるアイテム組合せの部分集合を抽出・列挙する問題

レコード番号	アイテム集合
1	a b c
2	a b
3	a b c
4	b c
5	a b
6	a b c
7	c
8	a b c
9	a b c
10	a b
11	b c

最小頻度 $\alpha = 10$

{ b }

最小頻度 $\alpha = 8$

{ ab, a, b, c }

最小頻度 $\alpha = 7$

{ ab, bc, a, b, c }

最小頻度 $\alpha = 5$

{ abc, ab, bc, ac, a, b, c }

最小頻度 $\alpha = 1$

{ abc, ab, bc, ac, a, b, c }



2012.08.09

9

LCM-ZDD 法による高速化

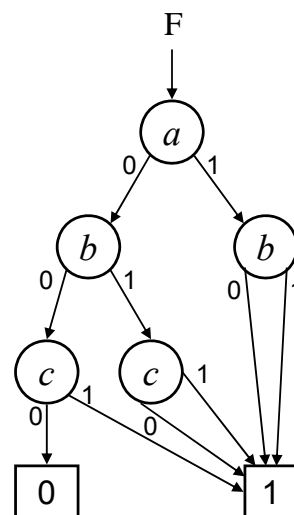
- 計算結果の頻出アイテム集合を、メモリ上に圧縮してZDDで表現し、そのポインタのみを返す。
 - 計算結果をファイルに出力しない。

レコード番号	アイテム集合
1	a b c
2	a b
3	a b c
4	b c
5	a b
6	a b c
7	c
8	a b c
9	a b c
10	a b
11	b c

LCM-ZDD法

(最小頻度 $\alpha = 7$)

{ ab, bc, a, b, c }



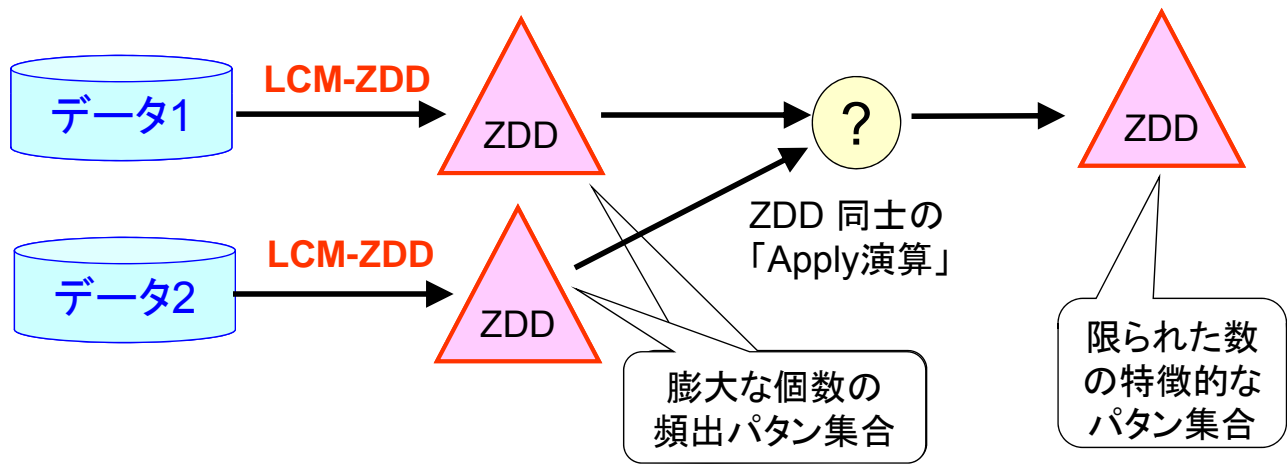
2012.08.09

10

Table 2. Comparison of LCM-ZDD with the previous LCM

Dataset name:	# solutions	CM of ZBDD	LCM-ZDD	CM-count	Original LCM	growth
min. support	itemssets	Time(s)	Time(s)	Time(s)	Time(s)	Time(s)
mushroom: 1,000	123,287	760	0.50	0.49	0.64	1.78
500	1,442,504	2,254	1.32	1.30	3.29	3.49
300	5,259,786	4,412	2.25	2.22	9.96	5.11
200	18,094,822	6,383	3.21	3.13	31.63	6.24
100	66,076,586	11,584	5.06	4.87	114.21	6.72
70	153,336,056	14,307	7.16	7.08	277.15	6.97
50	198,169,866	17,830	8.17	7.86	357.27	6.39
T10I4D100K: 100	27,533	8,482	0.85	0.85	0.86	209.82
50	53,386	16,872	0.97	0.92	0.98	242.31
20	129,876	58,413	1.13	1.08	1.20	290.78
10	411,366	173,422	1.55	1.36	1.64	332.22
5	1,923,260	628,491	2.86	2.08	3.54	370.54
3	6,169,854	1,576,184	5.20	3.15	8.14	386.72
2	19,561,715	3,270,977	9.68	5.09	22.66	384.60
BMS-WebView-1: 50	8,192	3,415	0.11	0.11	0.12	29.46
40	48,544	10,755	0.18	0.18	0.22	48.54
36	461,522	28,964	0.49	0.42	0.98	67.16
35	1,177,608	38,164	0.80	0.69	2.24	73.64
34	4,849,466	49,377	1.30	1.07	8.58	83.36
33	69,417,074	59,119	3.53	3.13	144.98	91.62
32	1,531,980,298	71,574	31.90	29.73	3,843.06	92.47
chess: 1,000	29,442,849	53,338	197.58	197.10	248.18	1,500.78
connect: 40,000	23,981,184	3,067	5.42	5.40	49.21	212.84
pumsb: 32,000	7,733,322	5,443	60.65	60.42	75.29	4,189.09
BMS-WebView-2: 5	26,946,004	353,091	4.84	3.62	51.28	118.01

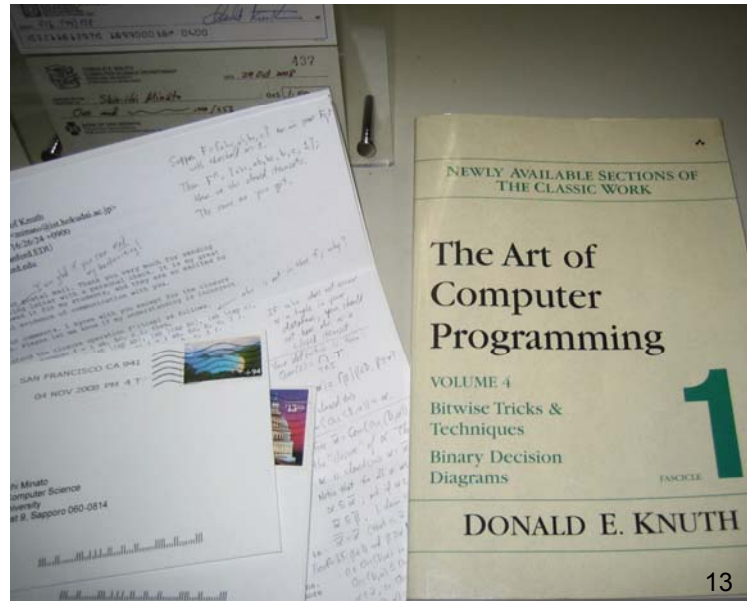
ZDDを用いたパタン集合演算



- 数十億ものパタンを含む集合を圧縮して表現し、ZDDの集合演算を使って効率よく絞込みを行える。
 - 従来の明示的な表現方法では、意味のある解析処理を現実的な時間で行うことは不可能

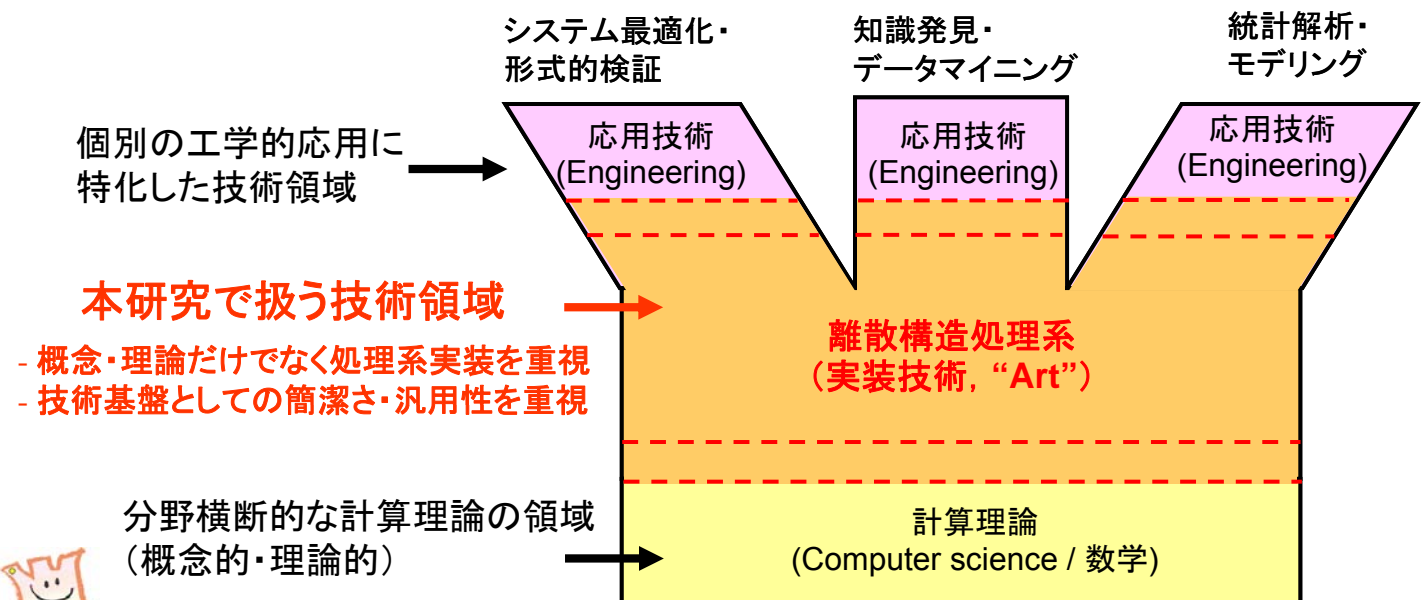
Knuthの名著とBDD/ZDD

- Knuthの世界的名著「The Art of Computer Programming」の最新巻 (Vol.4, Fascicle 1, 2009) で、BDDが取り上げられ、その中でZDDが30ページ以上に渡り詳しく解説された。
- **日本人の研究成果が、このシリーズに項目として詳細に掲載されるのは初めて。**
- Knuth氏本人から、ZDD考案者として校正作業への協力を依頼する長文のメールと手紙を受領。
- 2010年5月には湊がKnuth邸を訪問し、プロジェクトの方向性について意見交換を行った。



本研究プロジェクトの対象領域

BDD/ZDD技術の新しい切り口として、様々な離散構造を**統合的に演算処理**する技法を体系化し、**分野横断的かつ大規模な実問題**を高速に処理するための技術基盤を構築する。



研究実施場所とメンバ

■ 札幌メインオフィス(北大・情報科学研究科)

- ポスドク研究員6名
- 事務スタッフ3名、技術員2名、RA3名
- 北大研究者(GCOE他)と連携



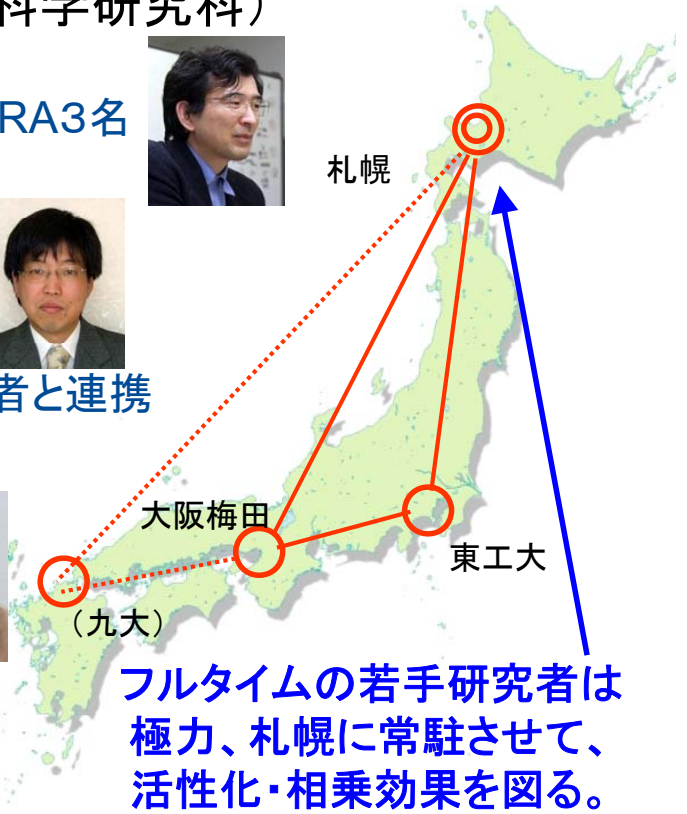
■ 東京サテラボ(東工大)

- GL:産総研・津田宏治氏
- ポスドク研究員2名、RA2名
- 東工大GCOE・東京地区研究者と連携



■ 大阪サテラボ(梅田駅北口)

- GL:阪大・鷺尾隆 教授
- ポスドク1名、技術員1名、RA1名
- 関西地区研究者と連携



各地区は高品質TV会議システムで常時接続



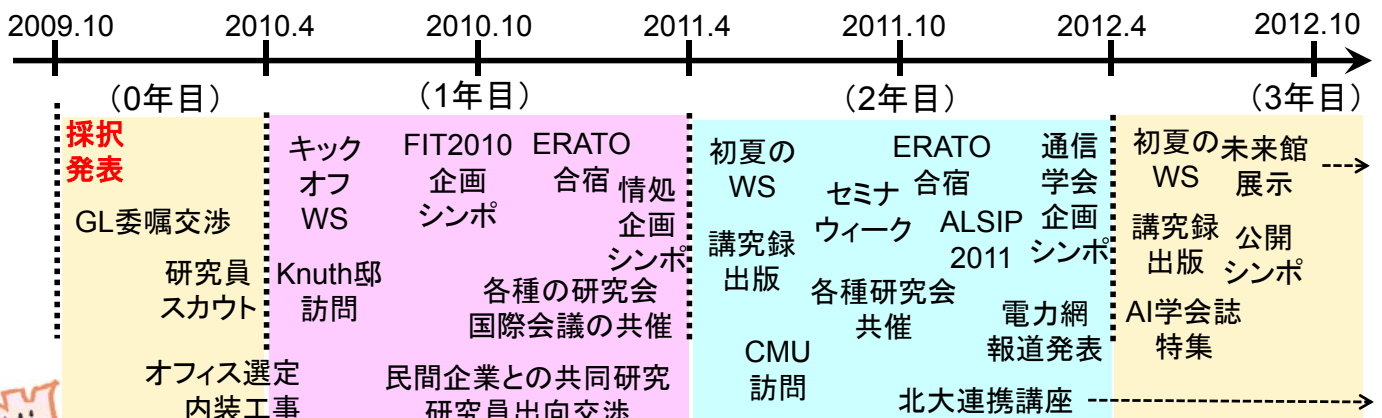
本ERATOプロジェクトのスケジュール

■ 研究活動も3年目に入る

- そろそろ折り返し点
- プレスリリースなど、それなりに分かりやすい成果は出ている。

■ 5年計画なので残り3年たらず

- ただし、4年目の年度末(今から2年たらず)に最終報告書を提出し、1年かけて評価を受ける。
- ソフトランディングで終了するか、別の大型プロジェクトに接続するか(?)



ZDDの応用

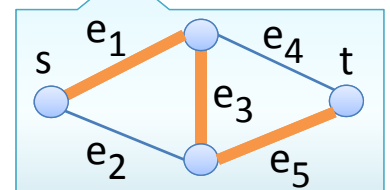
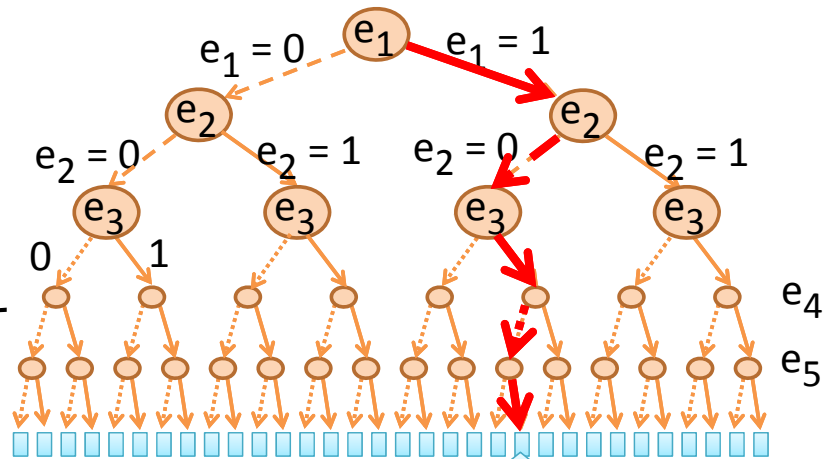
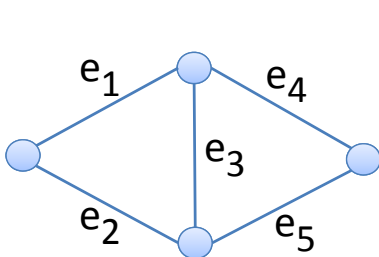
- 元々はLSI設計の論理式(CNF/DNF)の簡単化に利用
 - 膨大な項数の論理式を高速に因数分解する方法[Minato96]
 - 算術式の因数分解法[Minato97]
- データマイニングへの応用
 - ZDDを用いた頻出パターンマイニング[Minato2005]
 - LCM over ZDDs アルゴリズム[Minato-Uno2008]
 - 時分割データベースからのパターン変化の検出[Minato2010]
- グラフに関する種々の問題への応用
 - 最大クリーク問題、彩色問題、カバー問題等[Coudert97]
 - ZDDを用いた超高速パス列挙アルゴリズム[Knuth2009]
 - 電力ネットワークの制御、避難所配置問題、通信NW、他



2012.08.09

17

BDD/ZDDによるグラフの列挙



- 与えられた制約を満たすサブグラフ(枝の部分集合)を列挙する問題
 - 各枝の有無が入力変数となる
 - ZDD上での経路がサブグラフに対応
 - 制約を満たす:1-終端、満たさない:0-終端
- 制約条件を論理式で表せればApply演算でBDD/ZDDを生成できる

- 部分的に類似する組合せが多数発生する場合、圧縮率が高くなる
- ある枝を使うと決めたら別の枝が使えなくなる問題ではBDDよりも**ZDDが有利**



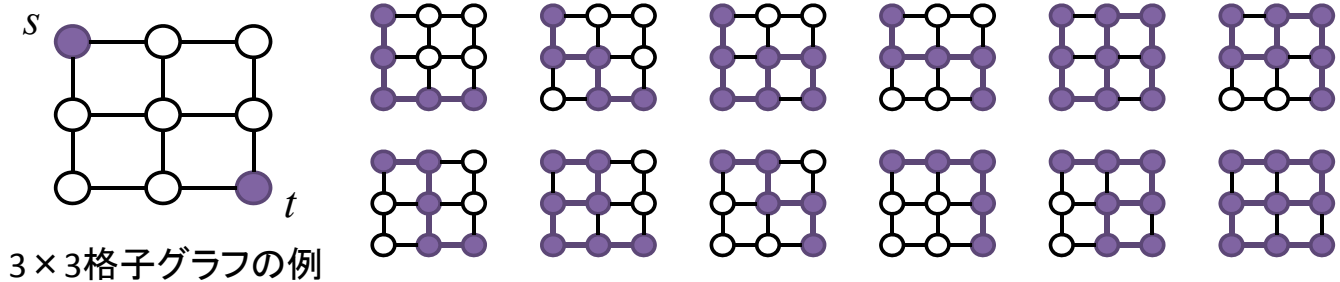
2012.08.09

18

ZDDを用いたパスの列挙 [Knuth2009]

- Knuth本のZDDの項目に演習問題として記載

- 入力: グラフ $G=(V, E)$, 2頂点 s, t
- 出力: 同じ頂点を2度通らない s から t までのパス (枝の組合せ) を列挙したZDD



応用は極めて多い

地理情報処理、ネットワークや論理回路の信頼性評価
電力網制御(スマートグリッド等)、ソフトウェア解析、etc.



2012.08.09

19

Knuthのパス列挙アルゴリズムSimpath

■ Knuth本人が実装したソースコードが公開されている。

- Texでコンパイルすると、Cコードとドキュメントを自動生成 (Latexではなく生Texを使用)
- 64ビット整数を使用できるように一部修正

■ 試してみると**驚くほど高速**。

15 × 15の2次元格子グラフ(枝の総数:420本)の
対角頂点を結ぶパスを全列挙するZDDの生成が可能に。

- 列挙したパスの総数:
227449714676812739631826459327989863387613323440通り
(→ 約 2.27×10^{47} 通り)
- 生成したZDDの節点数:
144,759,636 個(メモリ8GB以内)
- 計算時間:**わずか数分**



2012.08.09

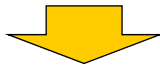
20

Simpathアルゴリズムの特徴

- Simpathは、ZDD同士の集合演算 (Apply演算) を使わず、与えられたグラフからパス列挙のZDDを直接構築

従来の常識:

問題専用のBDD/ZDD生成アルゴリズムを直接書くよりもApply演算を組合せて実装の方が工学的にエレガント



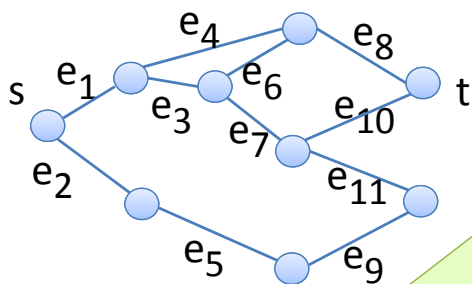
我々はApply演算だけを組合せてパス列挙のZDDを生成するアルゴリズムを作ってみたが、相当がんばって工夫してもSimpathが圧倒的に高速 (単なるコーディング技量の違いでは済まされない問題)



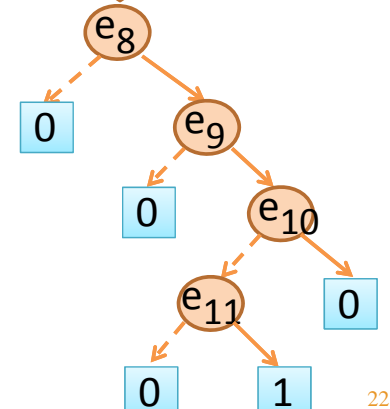
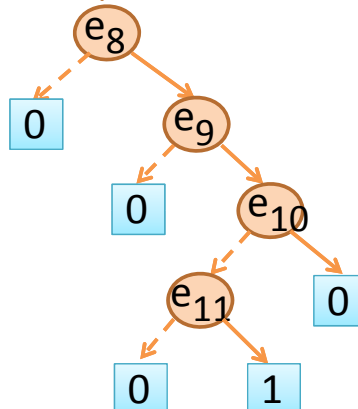
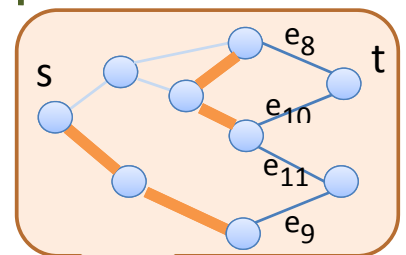
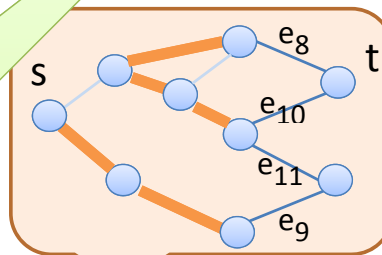
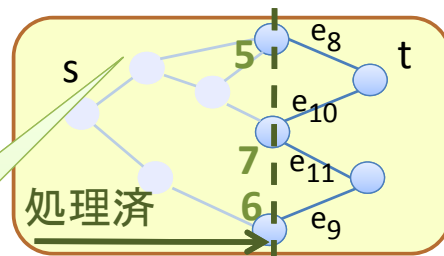
2012.08.09

21

Knuth によるアルゴリズム Simpath



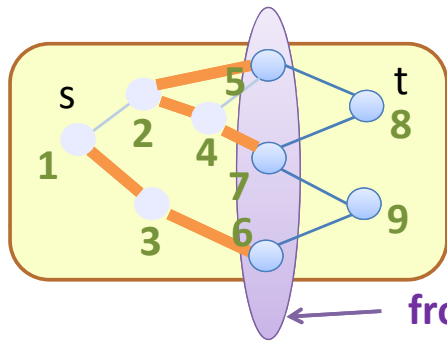
6はsとつながっている
5は7とつながっている



2012.08.09

Knuth によるアルゴリズム Simpath

接続情報の記憶法



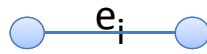
mate この部分のみ記憶

i	1	2	3	4	5	6	7	8	9
mate[i]	6	0	0	0	7	1	5	8	9

- 頂点がパスの端 → 逆端の番号
- 頂点がパスの途中 → 0
- 頂点がいずれのパスにも含まれない → 自身の番号

どの頂点の接続情報を覚えればよいか？

枝 i を処理するとき、
枝 i に接続している頂点を「訪れた」と表現する



$$F := \{\text{訪れたことのある頂点}\} \setminus \{\text{二度と訪れない頂点}\}$$

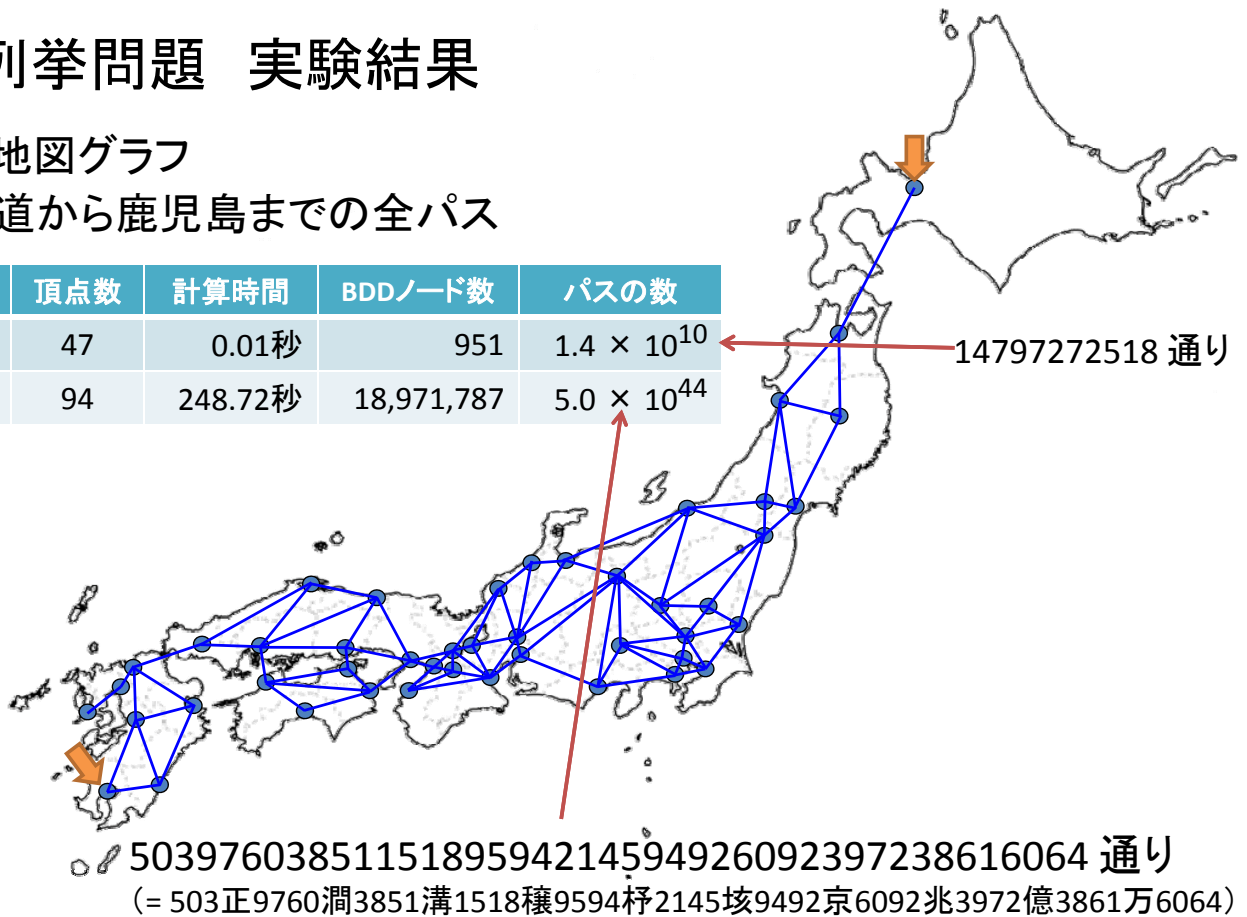
F をフロンティアという。フロンティア頂点の接続情報だけを憶える



パス列挙問題 実験結果

日本地図グラフ
北海道から鹿児島までの全パス

	頂点数	計算時間	BDDノード数	パスの数
日本地図	47	0.01秒	951	1.4×10^{10}
2重化	94	248.72秒	18,971,787	5.0×10^{44}

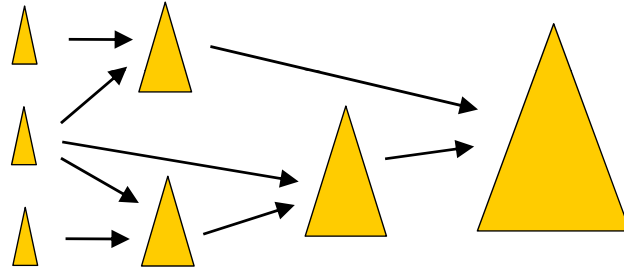


(= 503正9760潤3851溝1518穰9594杼2145垓9492京6092兆3972億3861万6064)

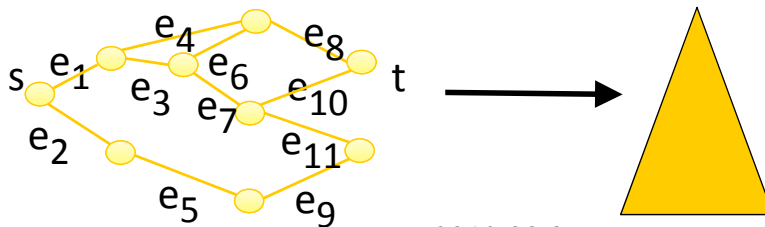


従来のZDD生成とSimpauth法の違い

- 従来法:
 - ZDD同士の集合演算を何度も適用し、所望のZDDを生成
 - 基本はBryantのApplyアルゴリズム



- Simpath法:
 - グラフをたどりながら、最終結果のZDDを一括生成
 - 問題に特有の性質を利用した動的計画法



2012.08.09

25



フロンティア法の一般化

- パス列挙のバリエーション
 - パス列挙 → サイクル列挙 (Knuth本に演習問題あり)
 - 無向グラフ → 有向グラフでも可能(これも演習問題)
 - 起点終点が複数ペアの場合でも可能(→非交差配線問題)
- ERATOで議論しているうちに、他にも様々なグラフ列挙問題に適用できることが分かってきた
 - 部分木列挙、大域木/林の列挙、カットセット列挙、グラフのk分割問題、連結確率の計算、完全/不完全マッチング列挙、etc.
- Tutte多項式を表すBDD生成法[Sekine-Imai95]との関係
 - さらに調べて行くと、90年代に東大・今井研で、Simpauthと極めて類似したアルゴリズムが研究されていたことがわかってきた。
 - ZDDではなくBDDを使用
 - パス列挙ではなく、連結成分の列挙
 - 途中状態として、mate情報ではなく、節点連結状態を保持
 - 「フロンティア」を「消去波面」と呼んでいるが本質的に同じ概念

2012.08.09

26



Tutte多項式

$$T(x, y) = \sum_{A \subseteq E} (x - 1)^{\rho(E) - \rho(A)} (y - 1)^{|A| - \rho(A)}$$

- グラフ $G(V, E)$ の不変量を表す多項式(参考文献[Welsh93])
 - $\rho(A)$ は、節点数 $|V|$ から枝集合 A の連結成分の個数を引いた数
 - 多項式を求めた後、 x, y に0や1などの適当な定数を代入すると、大域木の総数や、連結確率など、グラフの様々な性質がわかる。
- Tutte多項式の項数は、グラフサイズに対して指数爆発するので、単純に展開したのでは計算が困難
- 関根・今井らは、グラフをたどりながらTutte多項式を効率よく計算するためのBDD構築アルゴリズムを提案(1995頃)
 - BDDはTutte多項式をコンパクトに因数分解した形に対応
 - アルゴリズムはフロンティア法と極めて類似
 - 平面グラフや格子グラフに対する理論的計算量解析も提示
 - 当時は計算機性能が貧弱だったため、主に数学的興味に留まる

2012.08.09

27



フロンティア法の工学的応用

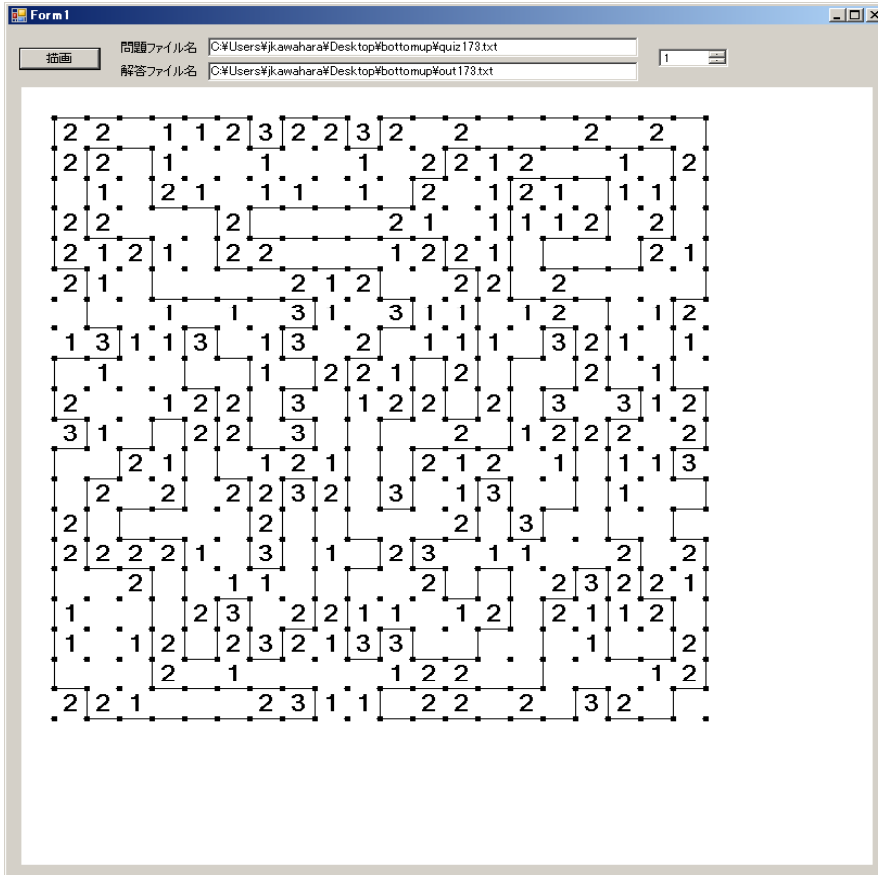
社会的に重要な様々な実問題に関係

- 有向/無向グラフのパス列挙:
 - 地理情報システム
 - 大規模システムの依存関係の解析、フローチャートの解析
 - ナンバーリンク、スリザーリンク等のパズル
 - 文字列の接続可能性の列挙
- グラフ k 分割問題:
 - 電力網の配電区割りの列挙
 - 避難所の配置問題
 - 選挙区割り問題
- 地理情報、電力網、物流網のような社会インフラの構造は、平面グラフや格子グラフに近い形であることが多い。
→ フロンティア法の効果が極めて高くなる傾向がある。

2012.08.09

28

条件付きパス(サイクル)の列挙



電力網への応用(研究成果2月プレスリリース)

- 林 泰弘 教授(早稲田大)との共同研究
 - 電力系スマートグリッド業界のリーダ的存在
(経産省スマートハウス標準化検討会座長、他多数の要職)
 - 電力網最適化の研究で1990年代より湊と協力関係
- 大震災後、より緊急性の高い研究課題に
 - 今後長期的に不足する電力を自然エネルギーで補うために必須の電力網解析・制御技術を支援

情報科学の研究者集団として
我が国の苦境を克服するため
できる限り貢献したい。

→ ERATOプロジェクト
での取り組みを加速

RIANT 先進グリッド技術研究所

Research Institute of Advanced Network Technolog

ホーム
トピックス
研究所案内
メンバー
アクセス

ホーム > 研究所案内

研究所案内

▶ 所長ご挨拶
▶ 先進グリッド技術研究所
▶ 研究テーマと分野
▶ 研究概要

所長ご挨拶

地球温暖化や環境破壊問題など地球規模の深刻な課題が取り上げられない日はありません。現代のエネルギー問題の解決には、この地球環境保全を考えつつエネルギーをいかに安定して供給するか、加えて経済成長をも調和させる取り組みが急務となっています。その究極の解決策になると期待されているのがスマートグリッドです。

スマートグリッドは電力エネルギーネットワークにITの技術や方法を応用していく、より効率的な電力利用の実現を目指します。太陽光発電や風力発電などの再生可能エネルギー電源の大量導入、ヒートポンプ給湯器などの熱エネルギー利用の電化、電気自動車の普及による移動手段の電化などを通して、CO2 排出量が少なくなる環境にやさしい電力社会、さらには量産型低炭素電力社会の到来に私たちの役割を担ってまいります。

林 泰弘 教授



電力網の問題

- どの変電所からどの領域に給電するか
 - 上手に切り替えれば損失を減らせる
 - 現状では家庭用太陽光発電が普及しても、電力網が不安定になるので、十分に活かさないで捨てている。これをうまく制御して活用できるようにしたい。
 - 災害故障の際にも配電網切り替えの問題が発生する。
- トポロジー制約と電氣的制約
 - 開閉器(スイッチ)のON/OFFで制御
 - 各領域はどこか1ヶ所の変電所から給電される。
 - 許容量以上に電流が流れない。
 - 末端電圧が低下しない。
 - できるだけ損失を抑えたい。

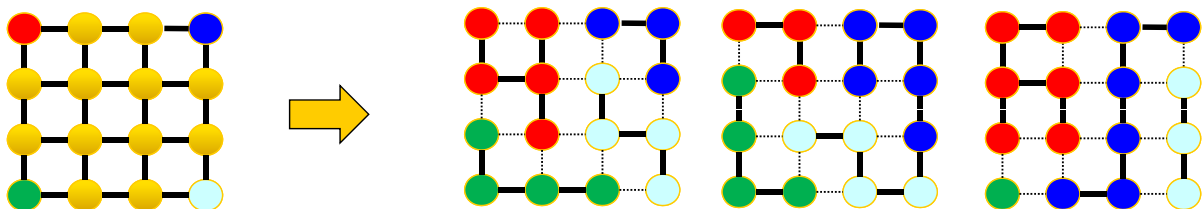
2012.08.09

31



グラフk分割問題

- グラフk分割問題(kカットセットの全列挙)
 - 与えられたk頂点を分離するカットセットを全列挙する問題
→ 全ての領域がどこか1つの電源から給電される
 - トポロジー制約についてはフロンティア法が使える



- 標準的な電力網モデル(スイッチ468個)で
トポロジ制約・電氣的制約を共に満たすZDDの生成に成功
ZDDノード数:約110万個(779MB) 実行時間:約1時間15分
解の個数:約 10^{63} (2136那由他8201阿僧祇3834恒河沙8532極9116載
8261正2214澗8049溝560穰9817杼8392垓4438京5235兆3981億8952万
1540)通り

2012.08.09

32



まとめ

- ZDDを用いた新しい列挙アルゴリズムの世界が広がり始めている
 - 単なる列挙ではなく索引化している
 - 単なる索引ではなくリッチな演算系(algebra)を備えている
 - 圧縮が効いている。
- 様々な実応用
 - パス列挙 → 地理情報処理、ソフトウェアのテスト
 - カットセット列挙 → 電力網、通信網、道路網、避難所配置問題
- 「最適化と列挙は車の両輪」(by 湊)
 - 最適化は世界中で競争が激しいが、列挙はまだそれほどでない
 - 日本は列挙の分野では、伝統的に強い
 - ZDDを利用する列挙・圧縮・索引化技術は、文句なく世界をリード

2012.08.09

33



日本科学未来館でのプロジェクト展示

- 日本科学未来館(東京・お台場)で我々の研究プロジェクトの展示「**フカシギの数え方**」を開催
 - 2012年8月1日～2013年2月25日(期間中、夏休み冬休み1回ずつ)
 - 小中高生・一般市民向けに研究内容をわかりやすく展示
- 8/25(土) 未来館ホールにてERATOプロジェクト2012公開シンポジウムを開催



@ Laboratory for New Media (Miraikan), The 11th Exhibition
The Art of 10⁶⁴ - Understanding Vastness -

JST ERATO Mianao Discrete Structure Manipulation System Project
JST ERATO 湊離散構造処理系プロジェクト

の	フ
数	カ
え	シ
方	ギ

@ Miraikan 第11期展覧会

2012.08.01 (wed.) ~ 2013.02.25 (mor.)