動的計画法を用いた有向二値完全系統樹の数え上げ

森戸　一貴，斎藤　寿樹，山口　一章，増田澄男
神戸大学大学院工学研究科電気電子工学専攻

種形質行列と有向二値完全系統樹

- **種形質行列**
  - すべての形質は二値
    - s 行 c 列の値が 1 ⇔ 種 s は形質 c を持つ
- **有向二値完全系統樹**
  - 各葉が1つの種のラベルを持つ順序なし根付き木
  - 各形質は一つの頂点にラベル付けされる
  - 種 s が形質 c を持つ
    ⇔ ラベル s を持つ葉はラベル c を持つ頂点の子孫である

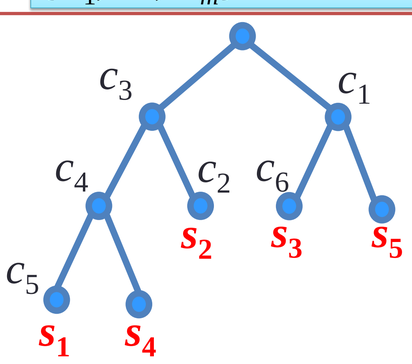
補題 [Jannson, 2008]

種形質行列Mが有向二値完全系統樹を持つ
⇔ 任意の二つの形質c_i, c_j について，次の3つのいずれかが成立
(i) $C_i \subseteq C_j$, (ii) $C_j \subseteq C_i$, (iii) $C_i \cap C_j = \varphi$

C_i : 形質c_iを持つ種の集合

$\{C_1, \ldots, C_m\}$がラミナー族

	c_1	c_2	c_3	c_4	c_5	c_6
s_1	0	0	1	1	1	0
s_2	0	1	1	0	0	0
s_3	1	0	0	0	0	1
s_4	0	0	1	1	0	0
s_5	1	0	0	0	0	0



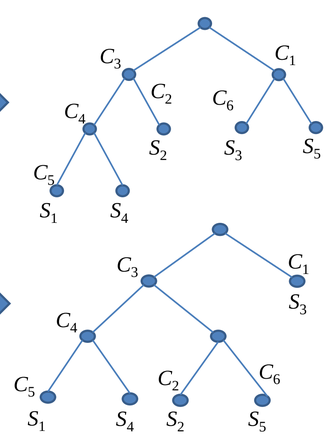
問題

入力: 不完全な種形質行列
　不完全：いくつかの要素が未知
出力: すべての有向二値完全系統樹
　すべての未知の要素に 0 もしくは 1 を割り当てる

	C_1	C_2	C_3	C_4	C_5	C_6
S_1	0	0	1	?	1	0
S_2	0	1	1	0	0	0
S_3	?	0	?	0	0	1
S_4	0	0	1	1	0	0
S_5	1	0	0	?	0	0

	C_1	C_2	C_3	C_4	C_5	C_6
S_1	0	0	1	1	1	0
S_2	0	1	1	0	0	0
S_3	1	0	0	0	0	1
S_4	0	0	1	1	0	0
S_5	1	0	0	0	0	0

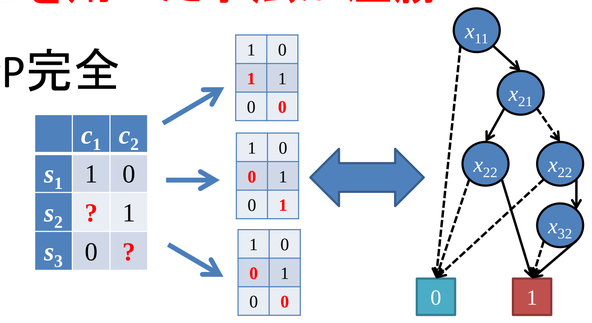
	C_1	C_2	C_3	C_4	C_5	C_6
S_1	0	0	1	1	1	0
S_2	0	1	1	0	0	0
S_3	0	0	1	0	0	1
S_4	0	0	1	1	0	0
S_5	1	0	0	0	0	0



既存研究

- Pe'er et al., 2004：
  - 系統樹を一つ求める多項式時間アルゴリズム
- Kiyomi et al., 2012
  - 系統樹の列挙アルゴリズム
    - 分枝限定法
    - ZDDを用いた手法
    } ZDDを用いた手法が圧勝
  - 系統樹の数え上げ問題：#P完全

	c_1	c_2
s_1	1	0
s_2	?	1
s_3	0	?



すべての解を保持するZDDを直接構築

提案アルゴリズム

●ノードの共有

次の二つのノードは共有できるか？

	C_1	C_2	C_3
S_1	1	1	0
S_2	0	1	0
S_3	0	0	1
S_4	?	0	?
S_5	1	?	?

	C_1	C_2	C_3
S_1	0	1	1
S_2	1	1	0
S_3	0	0	1
S_4	?	0	?
S_5	1	?	?

Yes!

すべての2列間の包含関係が等しい

$C_1 \subseteq C_2$, $C_1 \cap C_3 = \varphi$, $C_2 \cap C_3 = \varphi$

ラミナー配列にすべての2列の関係を保持

●ラミナー配列

すべての2列の関係もしくは関係の候補を格納

$$Laminar(C_i, C_j) = \begin{cases} Subset & C_i \subseteq C_j \text{ のとき} \\ Supset & C_j \subseteq C_i \text{ のとき} \\ Empty & C_i \cap C_j = \varphi \text{ のとき} \\ SubSup & C_i \subseteq C_j \text{ or } C_j \subseteq C_i \text{ のとき} \\ SubEmp & C_i \subseteq C_j \text{ or } C_i \cap C_j = \varphi \text{ のとき} \\ SupEmp & C_j \subseteq C_i \text{ or } C_i \cap C_j = \varphi \text{ のとき} \\ SubSupEmp & C_i \subseteq C_j \text{ or } C_j \subseteq C_i \text{ or } C_i \cap C_j = \varphi \text{ のとき} \end{cases}$$

	C_1	C_2	C_3	C_4	C_5	C_6
S_1	0	0	1	?	1	0
S_2	0	1	1	0	0	0
S_3	?	0	?	0	0	1
S_4	0	0	1	1	0	0
S_5	1	0	0	?	0	0

	C_1	C_2	C_3	C_4	C_5	C_6
C_1		Empty	Empty	Empty	Empty	SupEmp
C_2			Subset	Empty	Empty	Empty
C_3				Supset	Supset	SupEmp
C_4					SupEmp	Empty
C_5						Empty
C_6						

二つのラミナー配列が等しければ共有※

※正確には，割り当て中の行のすべての要素の値が等しいことも必要です．

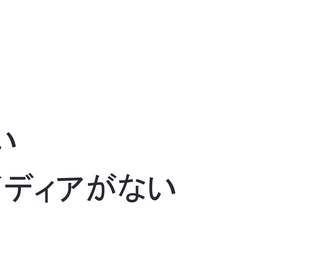
●計算時間

n：種の数
m：形質の数
D：未知の要素の数
d_{max}：未知の要素数が最も多い行の未知の要素数

- ラミナー配列の初期化：$O(nm^2)$時間
- ラミナー配列の更新：$O(m)$時間

各ノードが持つ情報
- ラミナー配列（サイズ:$1/2m(m\text{-}1)$）
- 探索中の行の未知の要素の値(高々d_{max})

各レベルでの部分問題の数は高々$7^{½\,m(m-1)} \cdot 2^{d_{max}}$個



全体の計算時間：
$nm^2 + Dm7^{½\,m(m-1)} \cdot 2^{d_{max}}$

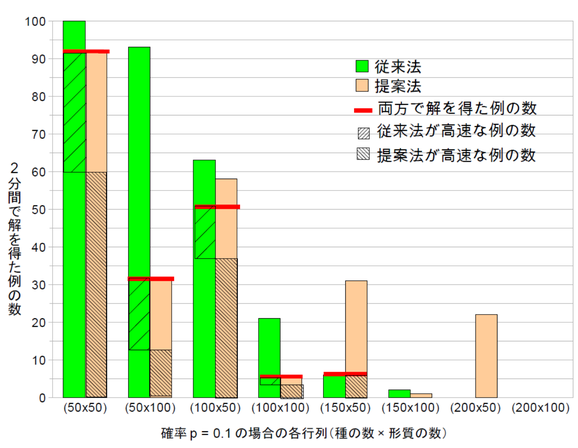
計算機実験

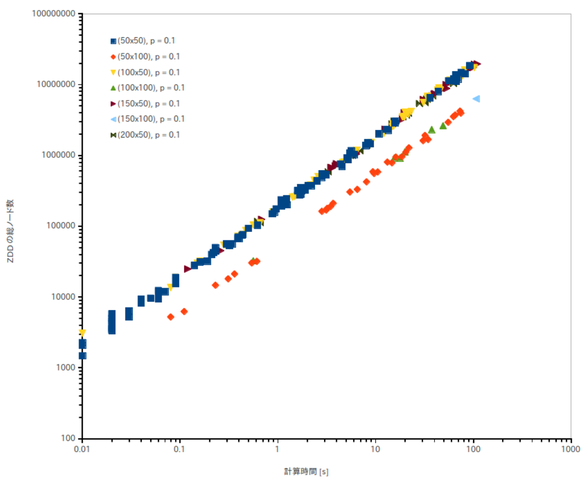
実験データ
- ランダムな種形質行列を生成
- 各要素を確率0.1で未知の要素に書き換え
  - 種の数：50, 100, 150, 200
  - 形質の数：50, 100

それぞれ100個生成

実験環境
- CPU: Core i3 − 2120
- Memory: 16GB
- ZDDライブラリ：SAPPOROBDD

タイムアウト2分





まとめと今後の課題

- まとめ
  - 有向二値完全系統樹の数え上げアルゴリズムを開発
    - ZDDの直接構築
    - 種の数に依存しない
    - 形質の数が少ないときは従来法よりも高速
- 今後の課題
  - より良い共有化の提案
  - 変数順序の問題
    - 現在の変数順序はあまりよくない
    - よい変数順序では共有化のアイディアがない
  - 実データを使った実験