Efficient Enumeration of the Directed Binary Perfect Phylogenies from Incomplete Data

Toshiki Saitoh (Kobe University)

Joint work with

Masashi Kiyomi (Yokohama City University)

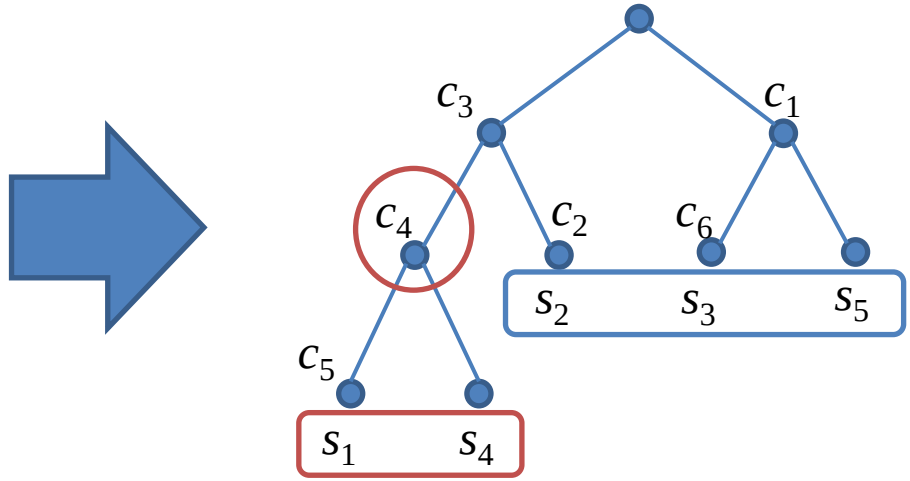
Yoshio Okamoto (The University of Electro-Communications)

Directed Binary Perfect Phylogeny

- **Input:** A species-character matrix M
  - All characters are binary.
    - $m_{sc} = 1$ iff the species s has the character c
- **Output:** A directed perfect phylogeny
  - An unordered rooted tree whose leaves have one species label.
  - Each character is labeled one node.
  - A species s has a character c **if and only if** the leaf with label s is a descendant of the node with label c.

○ 0 → 1
× 1 → 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



Directed Binary Perfect Phylogeny

Lemma [Jannson, 2008]

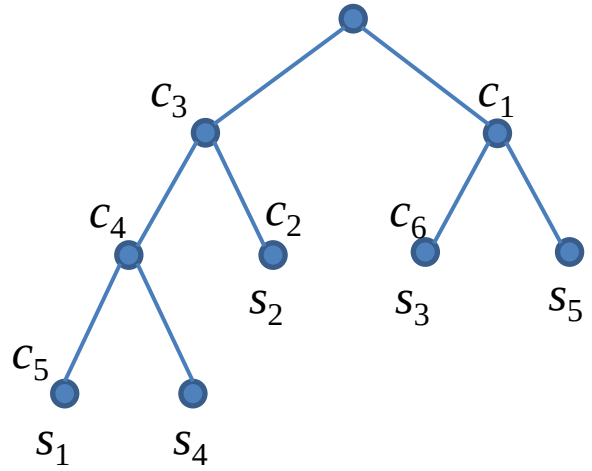
A matrix M admits a directed perfect phylogeny **if and only if** for every pair of columns i and j, either C_i and C_j are disjoint or one contains the other.

C_i : the set of species with the character c_i

We can construct a phylogeny in polynomial time.

$C_3=\{s_1, s_2, s_4\}$ $C_4=\{s_1, s_4\}$ $C_6=\{s_3\}$

	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



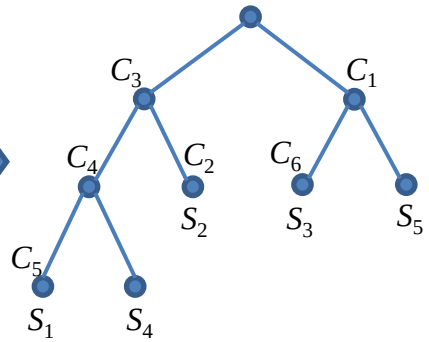
$C_4 \subseteq C_3$ $C_3 \cap C_6 = \emptyset$

Incomplete Directed Perfect Phylogenies

- **Input:** An incomplete species-character matrix
  - The states of some characters are unknown.
- **Output:** A directed perfect phylogeny
  - The unknown states are completed

	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



We can find one phylogeny in polynomial time.

[Pe'er et al., 2004]

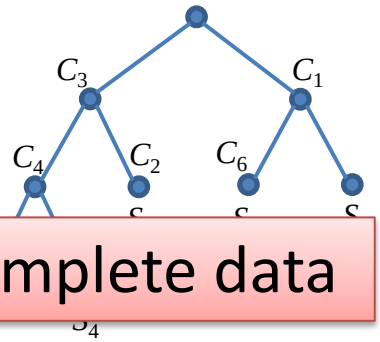
Incomplete Directed Perfect Phylogenies

- **Input:** An incomplete species-character matrix
  - The states of some characters are unknown.
- **Output:** A directed perfect phylogeny
  - The unknown states are completed

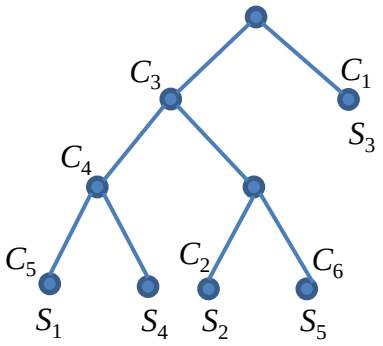
	C_1	C_2	C_3	C_4	C_5	C_6
S_1	0	0	1	?	1	0
S_2	0	1	1	0	0	0
S_3	?	0	?	0	0	1
S_4	0	0	1	1	0	0
S_5	1	0	0	?	0	0

Enumeration of all perfect phylogenies from incomplete data

	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

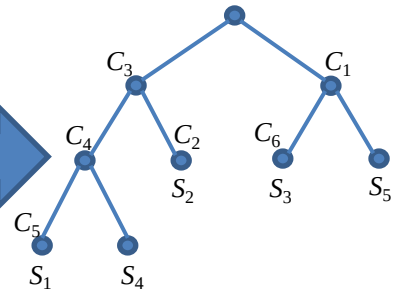


Why Enumeration?

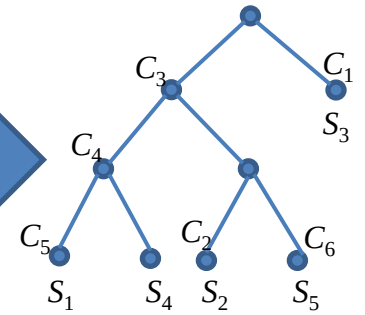
- Data mining
  - Extraction of characters from all objects
- Indexing
  - Counting
  - Random sampling
  - Searching
  - Filtering

	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



Our Contribution

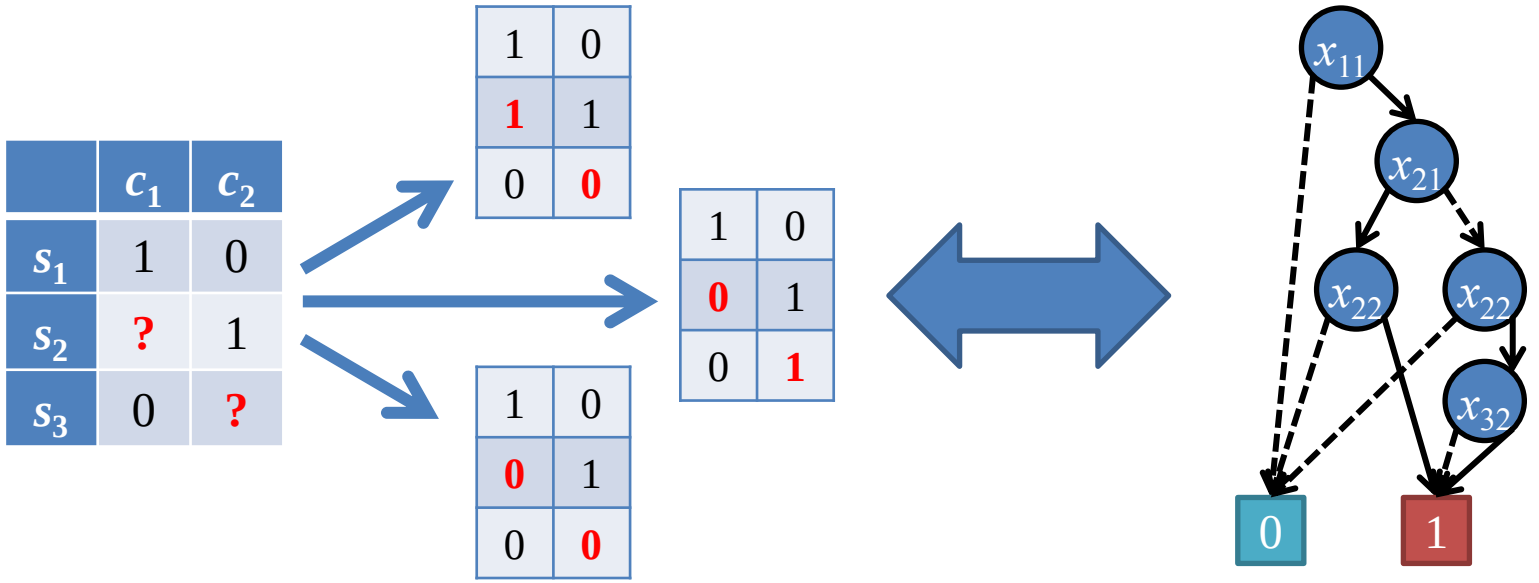
- Proposing two enumeration algorithms
  - Branch and bound (B&B)
    - Output all perfect phylogenies one by one
    - Runs in O($|M|\ k\ h$) time
      - k: #"?" in M, h: #perfect phylogenies
  - ZDD approach
    - Represent all perfect phylogenies compactly
    - Many applications
      - Counting, random sampling, filtering
- Proof of #P-hardness of the counting problem
  - Reducing by counting the number of matchings in a bipartite graphs

Perfect Phylogenies and ZDD

- Introducing a boolean variable x_{sc} for each species s and character c
  - x_{sc} = 1 if and only if the species s has the character c.

Lemma [Jannson, 2008]

A matrix M admits a directed perfect phylogeny **if and only if** for every pair of columns i and j, either C_i and C_j are disjoint or one contains the other.



Perfect Phylogenies and ZDD

- Introducing a boolean variable x_{sc} for each species s and character c
  - x_{sc} = 1 if and only if the species s has the character c.

Lemma [Jannson, 2008]

A matrix M admits a directed perfect phylogeny **if and only if** for every pair of columns i and j, either C_i and C_j are disjoint or one contains the other.

for every distinct character c_i and c_j

exactly one of the following three is satisfied.

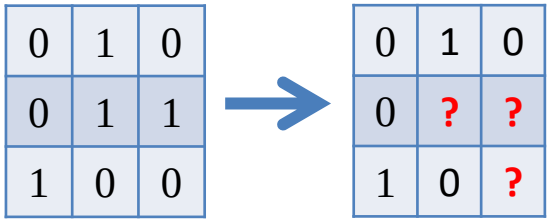
A) **for all** species s, **if** $x_{sc_i}=1$ **then** $x_{sc_j}=1$ → $C_i \subset C_j$

B) **for all** species s, **if** $x_{sc_i}=0$ **then** $x_{sc_j}=0$ → $C_j \subset C_i$

C) **for all** species s, **if** $x_{sc_i}=1$ **then** $x_{sc_j}=0$ → $C_i \cap C_j = \emptyset$

Experiments

- Instances:
  - Constructing an incomplete data from complete data
    - Random data set [Hudson, 2002]
    - "1" or "0" -> "?" with probability p (={0.1, 0.2, 0.3, 0.4, 0.5})
  - Matrix size (n, m): ({50, 100}, {50, 100})
  - 100 instances for each triple (n, m, p)



- B&B algorithm is written by C.
- ZDD approach is written by C++ (ZDD library is developed by Minato)

- Machine spec
  - OS: SuSE Linux Enterprise Server 10
  - CPU: Quad-Core AMD Opteron Processor 8393
    - #CPUs 16, #Processors 32, Clock Freq. 3092MHz
  - Memory: 512GB

Experimental Results

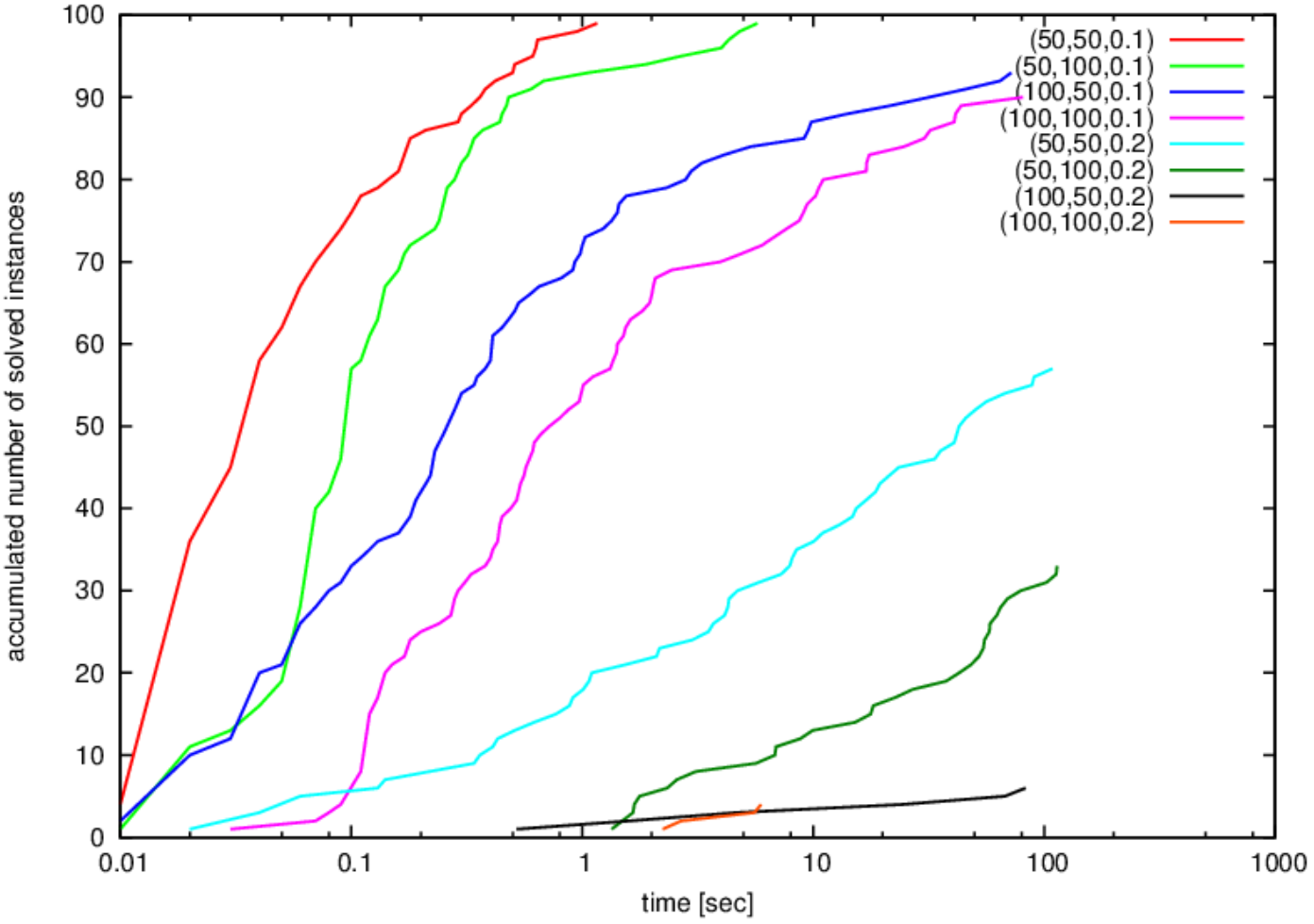
The number of solved instances by B&B and ZDD approach.

("solved" means that the algorithm successfully halts.)

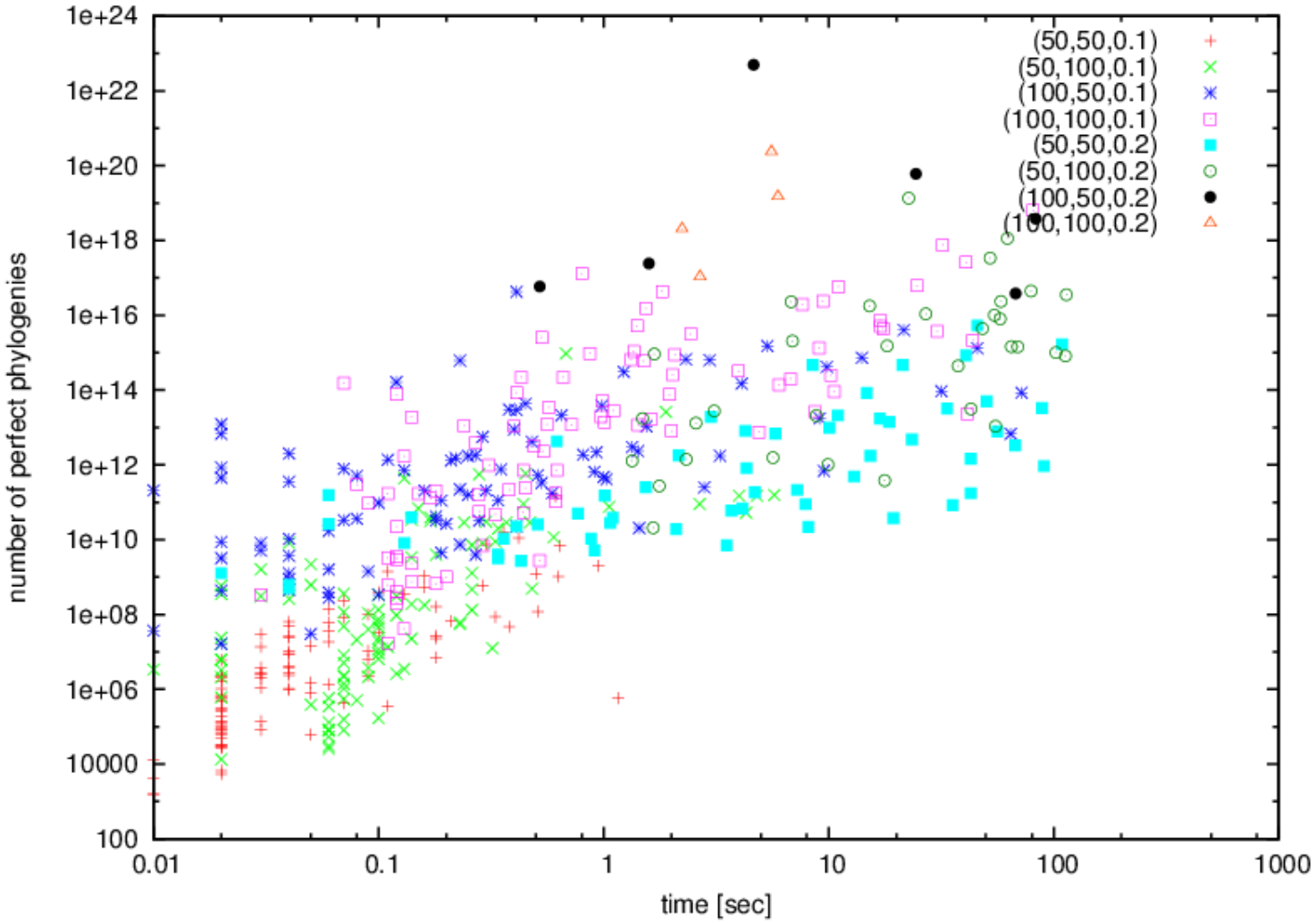
	B&B				ZDD Approach			
m, n	50, 50	50, 100	100, 50	100, 100	50, 50	50, 100	100, 50	100, 100
p=0.1	52	17	0	0	99	99	93	90
p=0.2	0	0	0	0	57	33	6	4

Timeout: 2 minutes

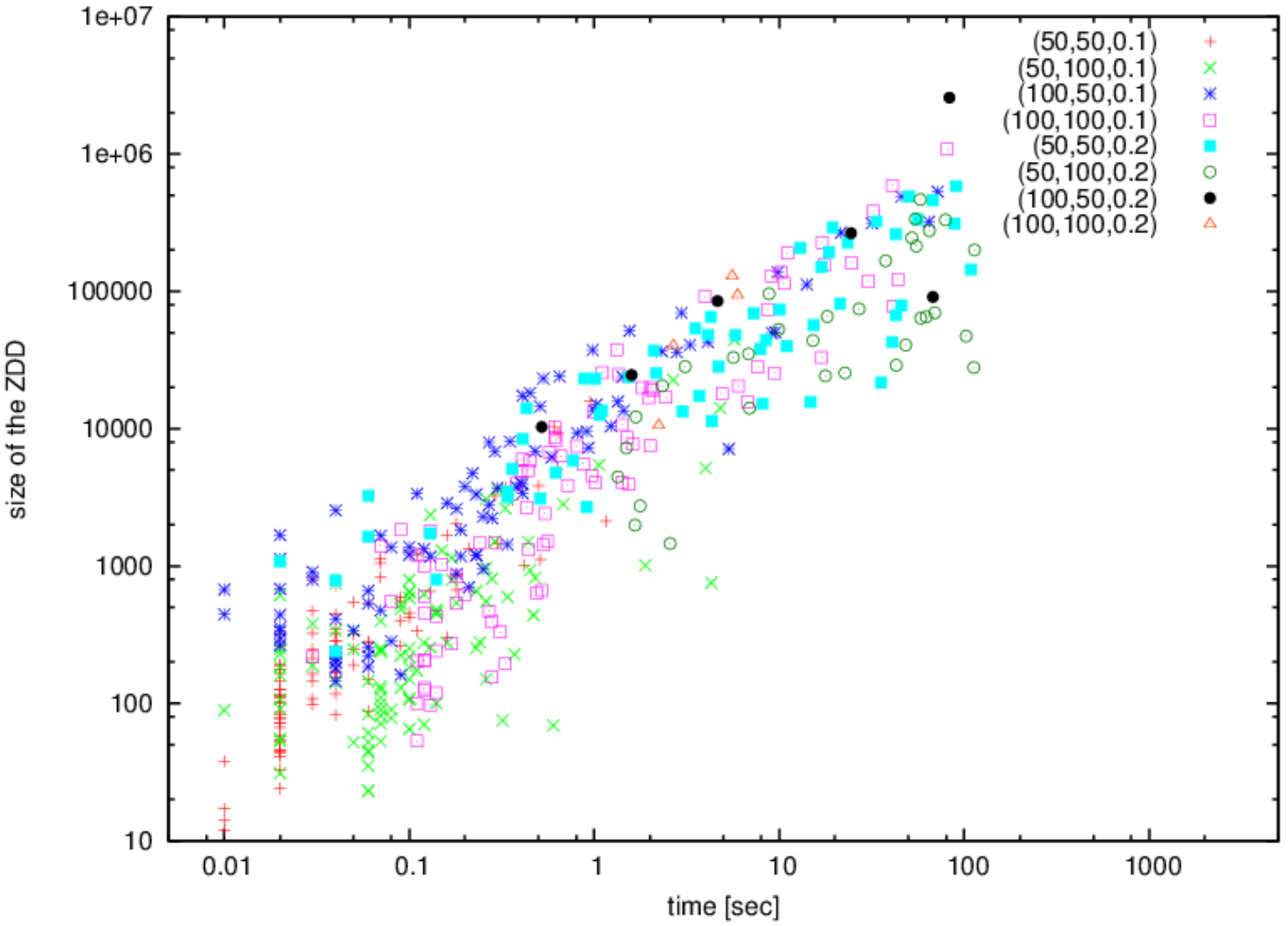
Experimental Results



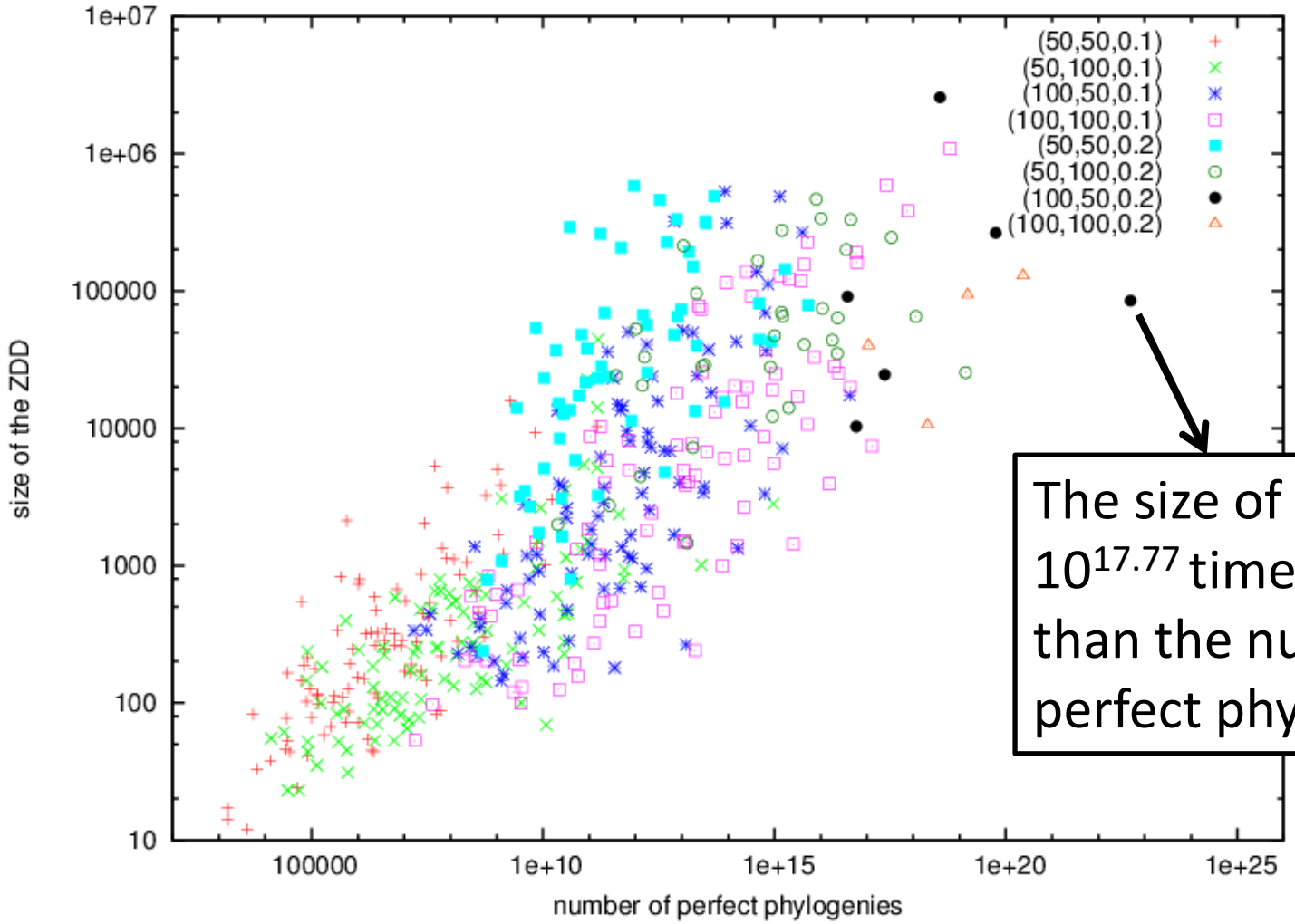
Experimental Results



Experimental Results

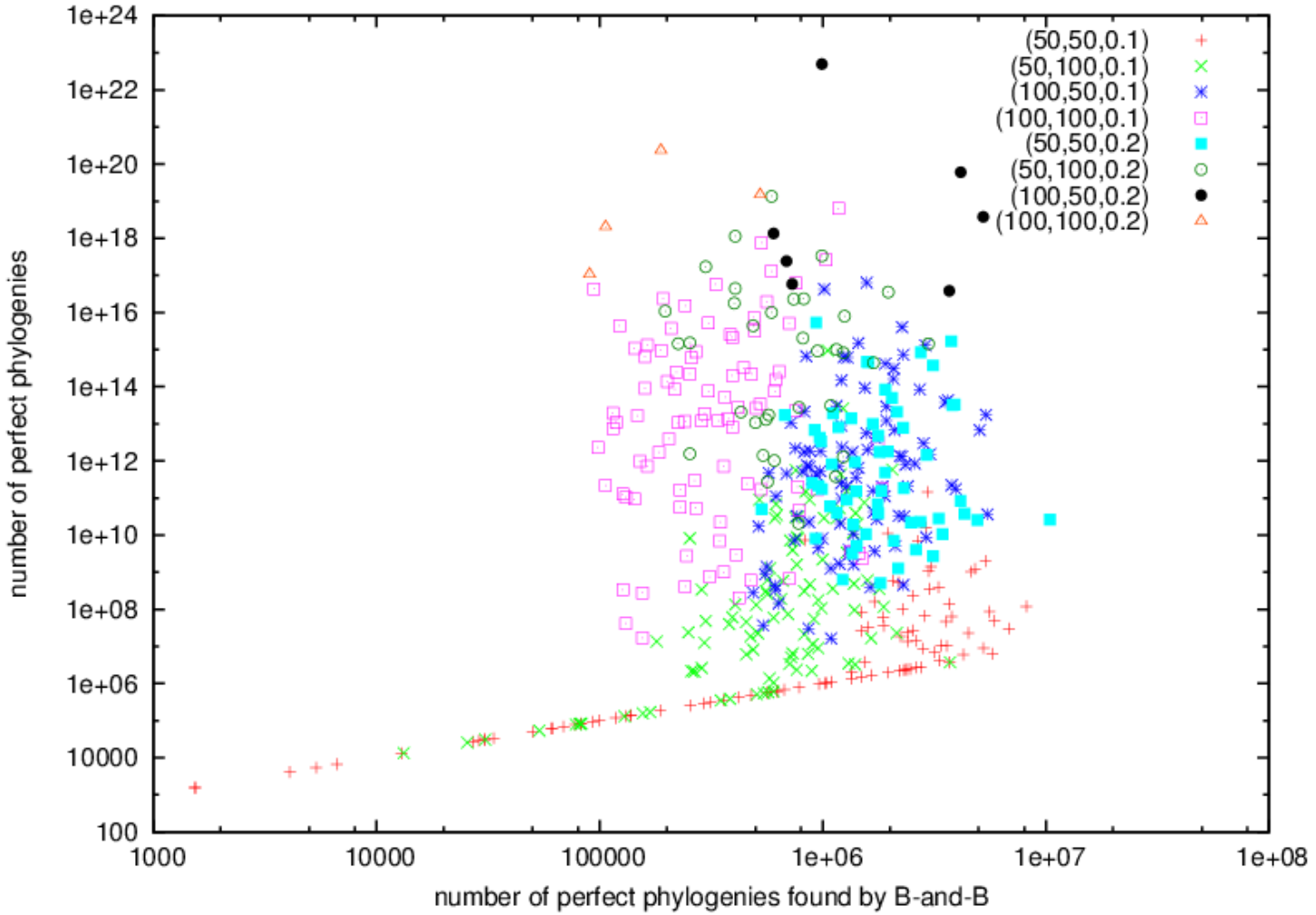


Experimental Results

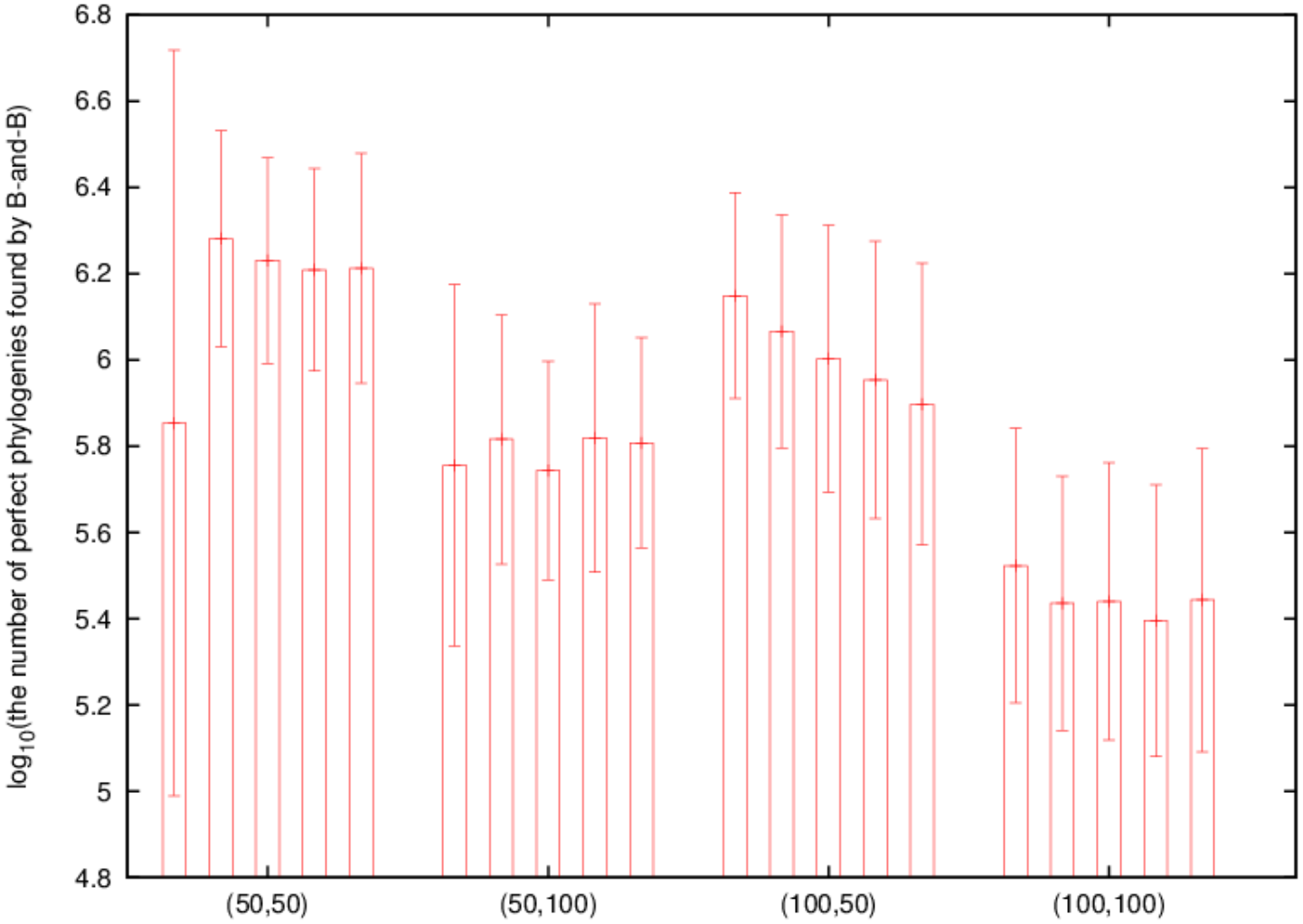


The size of ZDD is $10^{17.77}$ times smaller than the number of perfect phylogenies.

Experimental Results (B&B)



Experimental Results (B&B)



Conclusion

- Our results
  - Proposing two enumeration algorithms
    - Branch and bound algorithm (B&B)
    - ZDD approach
      - ZDD approach solved more instances than B&B.
      - Spends more time with more the ZDD size.
      - Show high compression rate of ZDD for the random data.
  - Proof of #P-hardness of the counting problem

Thank you for your attention!

Formulation

1. **for every** s, c **if** $m_{sc}=1$ **then** $x_{sc}=1$
2. **for every** s, c **if** $m_{sc}=0$ **then** $x_{sc}=0$
3. **for every** distinct c_i, c_j

   exactly one of the folowing three is satisfied.

   A) **for all** s, **if** $x_{sc_i}=1$ **then** $x_{sc_j}=1$ → $C_i \subseteq C_j$

   B) **for all** s, **if** $x_{sc_i}=0$ **then** $x_{sc_j}=0$ → $C_j \subseteq C_i$

   C) **for all** s, **if** $x_{sc_i}=1$ **then** $x_{sc_j}=0$ → $C_j \cap C_i = \varphi$

Step 3: for every distinct ci, cj

3.1 : ga = gb = gc = F;

3.2A: for all s,

ga = ga / xsci / xscj * xscj * xsci ∨ ga % xsci

3.2B: for all s,

gb = gb % xsci % xscj ∨ gb / xsci * xsci

3.2C: for all s,

gc = gc / xsci % xscj * xsci ∨ gc % xsci

3.3: F = ga ∨ gb ∨ gc