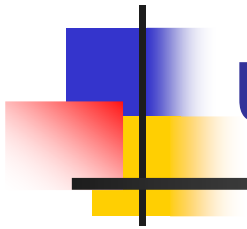


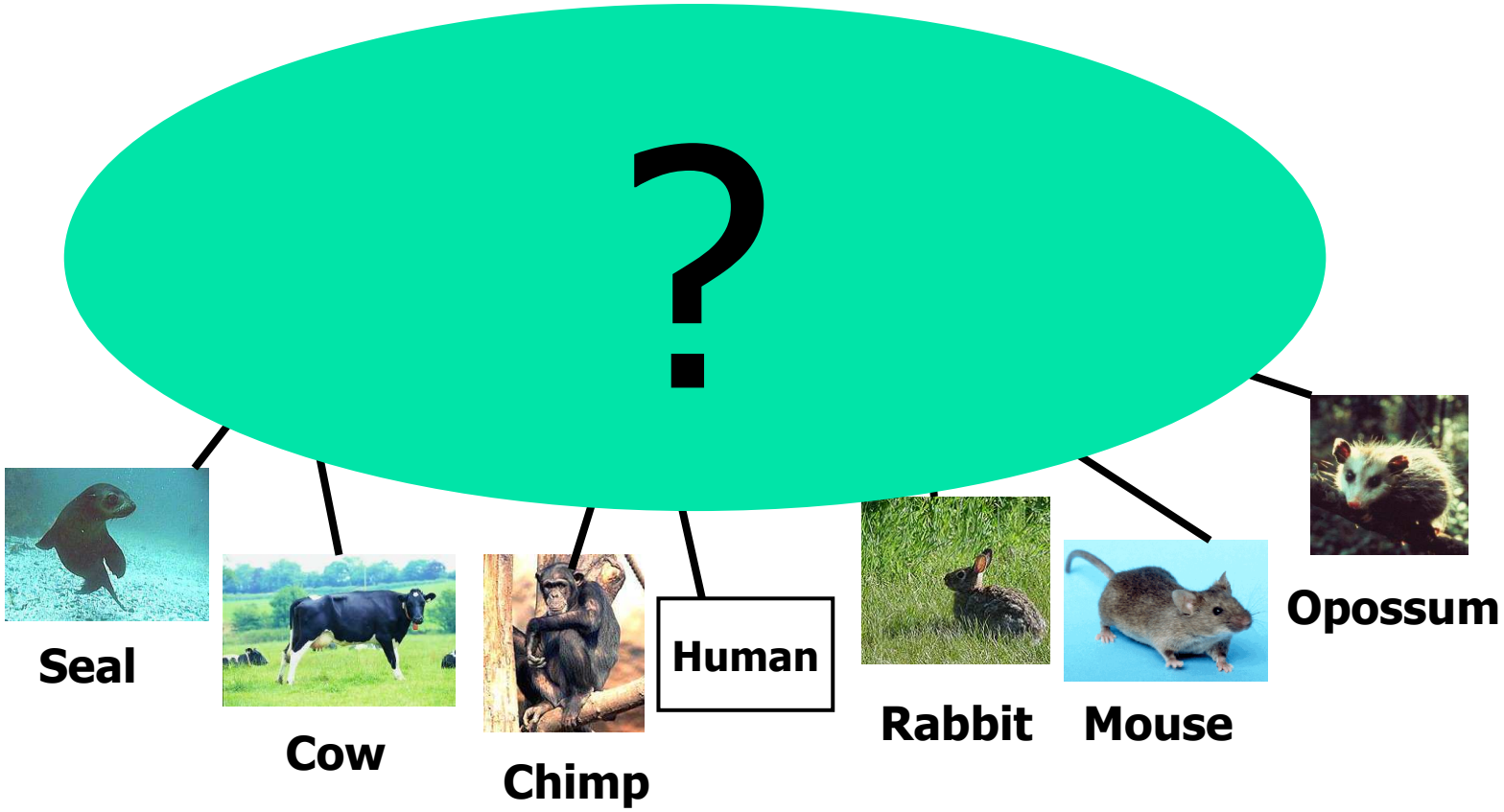
Parallelization of Phylogenetic Tree Inference using Grid Technologies

Yo Yamamoto[1]
Hidetoshi Shimodaira[1]
Hidemoto Nakada[1,2]
Satoshi Matsuoka[1,3]

1. Tokyo Institute of Technology
2. National Institute of Advanced Industrial Science and Technology
3. National Institute of Informatics

What is Phylogenetic Tree?

- A tree represents “evolution path” of life forms
  - How did they evolve from “the common ancestor”

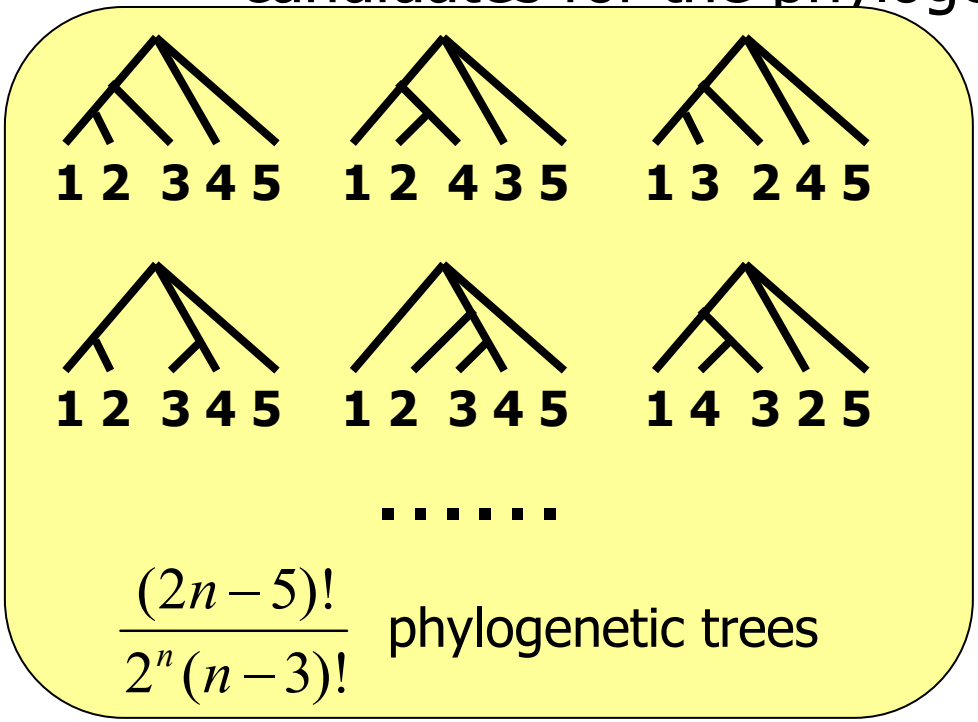


How to “guess” the tree

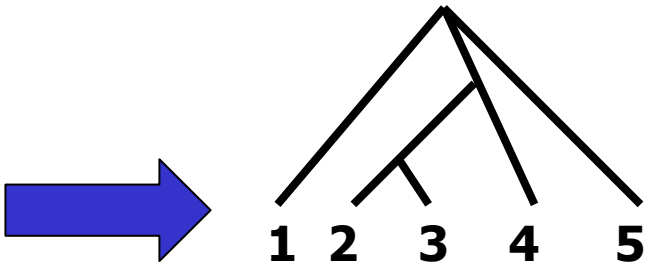
- Traditionally, guessed based on shapes
- Nowadays, we can guess based on DNA sequences
  - Define a measure called “likelihood” and compute it on every possible phylogenetic trees

Phylogenetic tree inference

- Look for the phylogenetic tree that gives the largest likelihood
  - Compute likelihood values for every phylogenetic trees
  - Practically, take few best phylogenetic trees as candidates for the phylogenetic tree



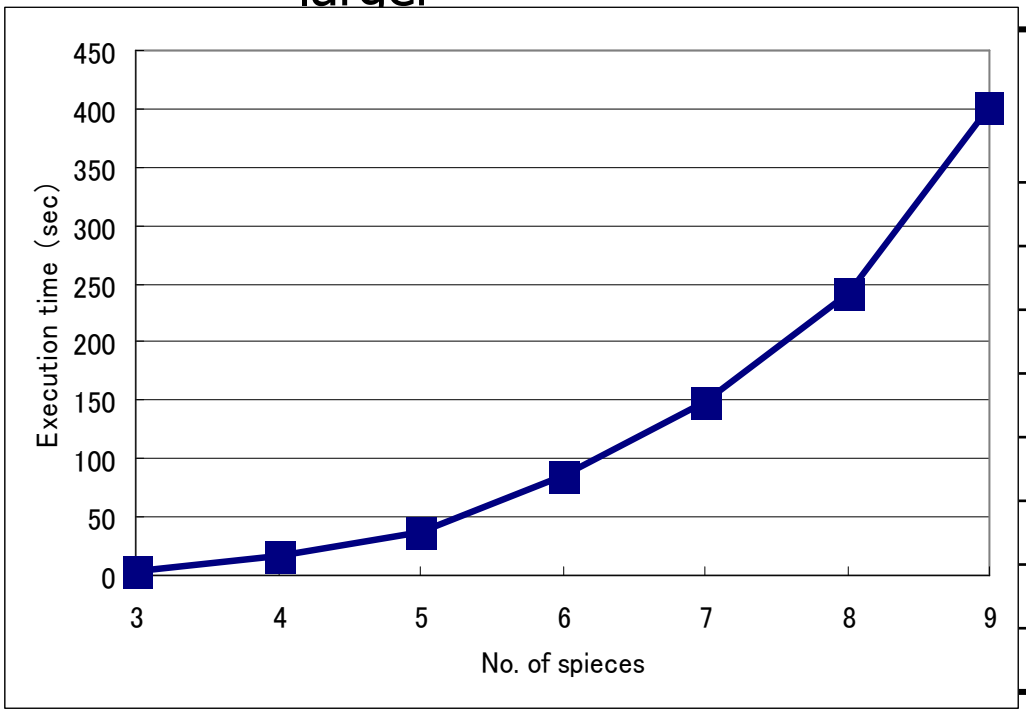
$$\frac{(2n-5)!}{2^n(n-3)!}$$ phylogenetic trees



Take a tree that gives
Largest likelihood

Problem: Huge computation!

- Use maximum likelihood method (paml)
  - phylogenetic tree inference cost = likelihood computation × no. of phylogenetic trees
  - For n species, no. of phylogenetic tree = $O(2^n n!)$
  - Computation time for a phylogenetic tree also gets longer as n get larger



Average Execution time for Maximum likelihood method (paml)

No.of species	No. of phylogenetic tree	Average execution time(min)	Predicted Execution time
3	1	4	4 sec
4	3	16	48 sec
5	15	37	9min 30sec
6	105	85	2hour30min
7	945	149	1day15hour
8	10395	241	29 days
9	135135	330	1.4year
10	2027025	418	27.3 years

Evaluation environment：
A node of abacus cluster

To cope the problem

- A approximate method called ‘split decomposition [shimodaira 01]’ is proposed by one of the authors
- This method drastically reduces computation cost for each tree
- But
  - The number of the tree still too big

Goal of this work

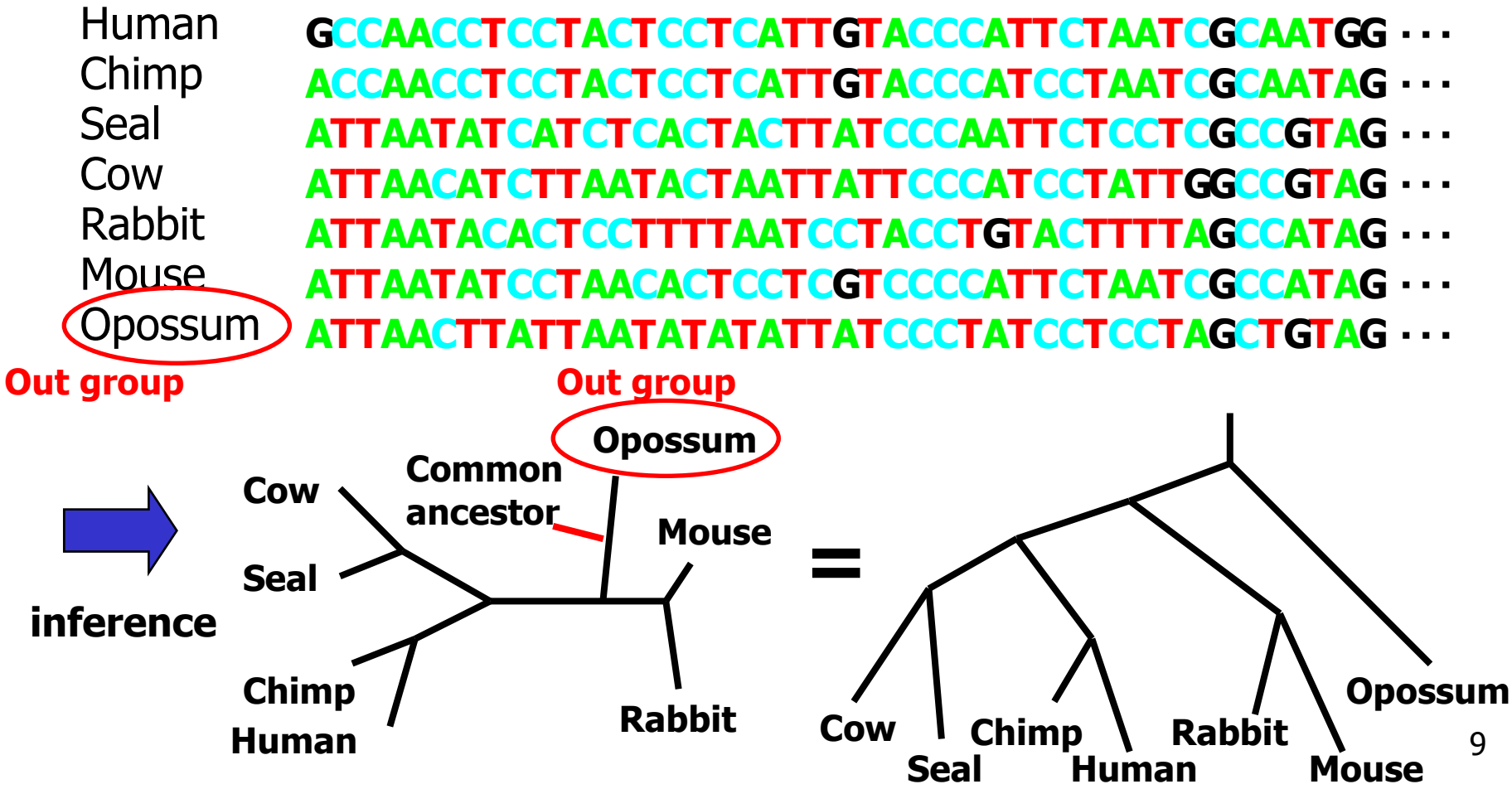
- Apply combinatorial optimization techniques to reduce the trees to be computed
  - Branch and Bound Method
  - Simulated Annealing
- Speed it up by parallelizing them using Grid middlewares – Ninf and Jojo
  - Likelihood computation
  - Combinatorial Optimization

Outline

- Phylogenetic tree and likelihood
- Split decomposition
- Overview of our system
  - Branch and bound
- Evaluation
- Conclusion

Phylogenetic Tree Inference

- Infer phylogenetic tree using DNA sequences

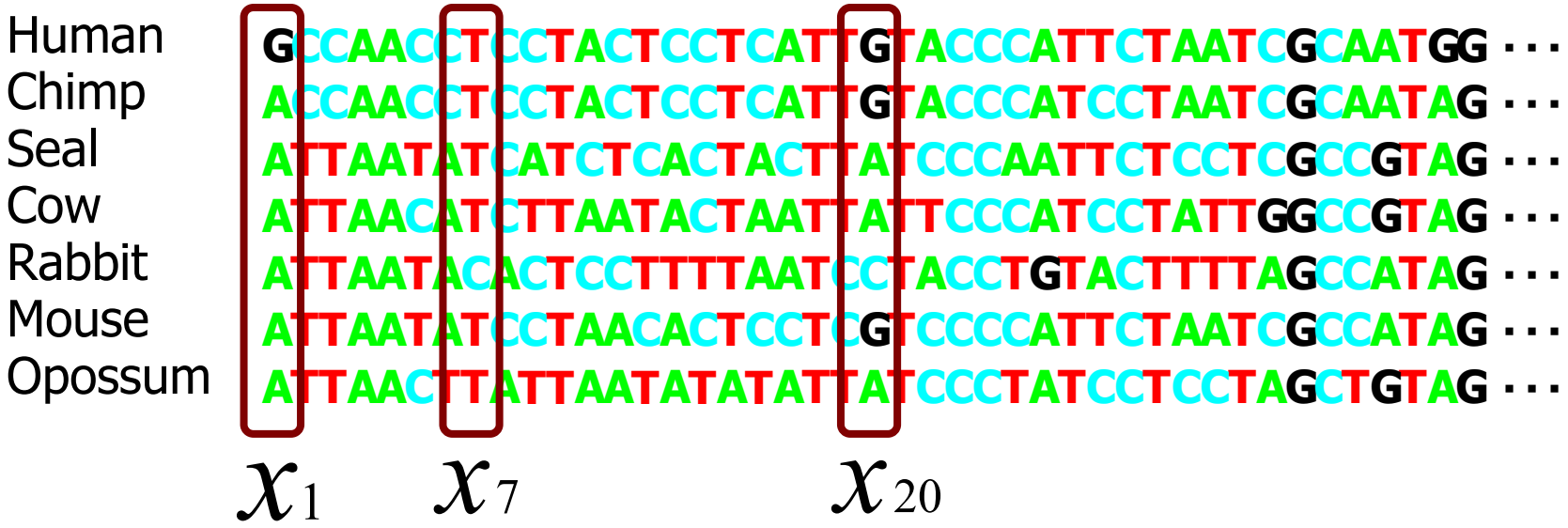


Phylogenetic tree likelihood

- Likelihood for a DNA sequence can be obtained as the product of Likelihood for each locus

$$L(p) = L(x_1) \bullet L(x_2) \bullet \Lambda \bullet L(x_m)$$

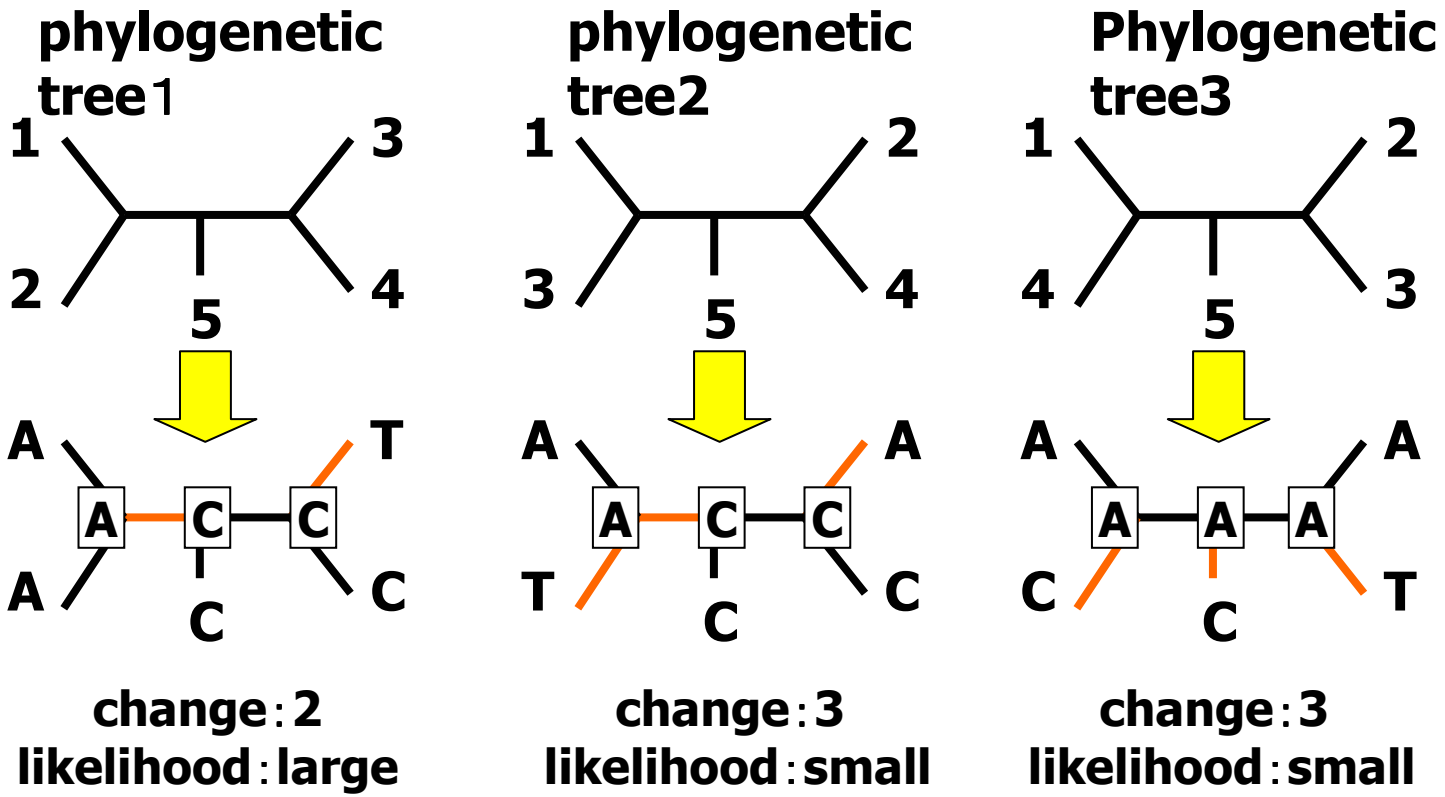
DNA sequence length: m = thousands to billions



Likelihood for a locus

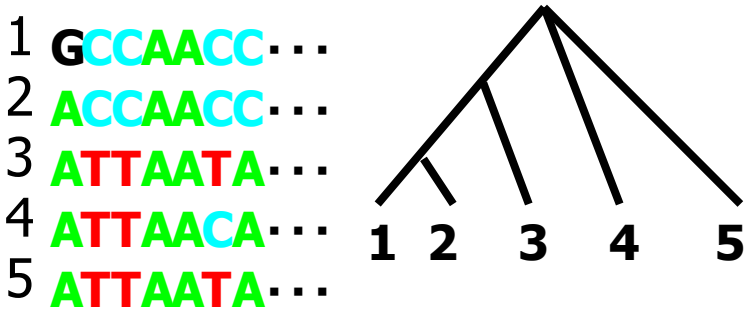
- Likelihood — how many times a entry changed
  - Smaller times – large likelihood

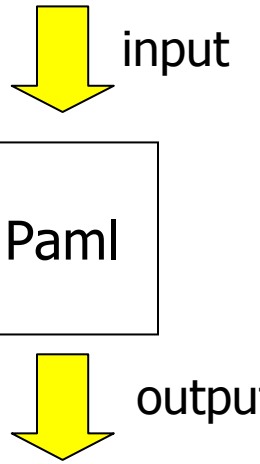
locus $x_k = \{A, A, T, C, C\}$



Programs to compute likelihood

- Paml [Yang, ’97]
  - Just compute likelihood for given phylogenetic tree
  - No-search method provided
  - Base-sequence, amino-acid sequence
- Molphy [Adachi et al., ’96]
  - Base, amino-acid sequence
  - Provides heuristics search
- Phylip [Shurman et al., ’80]
  - Base, amino-acid, protein sequence
  - Provide several search method

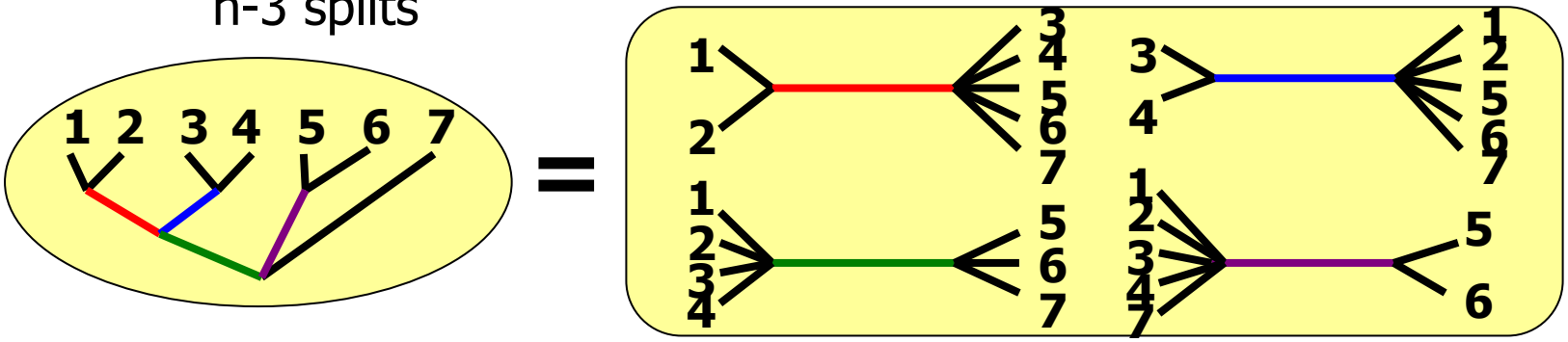




Logarithmic likelihood
lnL= - 912.5

Split decomposition

- Splits – components for a phylogenetic tree
  - Each split represents a branch of a phylogenetic tree
  - A phylogenetic tree for n species can be decomposed into n-3 splits



A phylogenetic tree **n-3 splits**

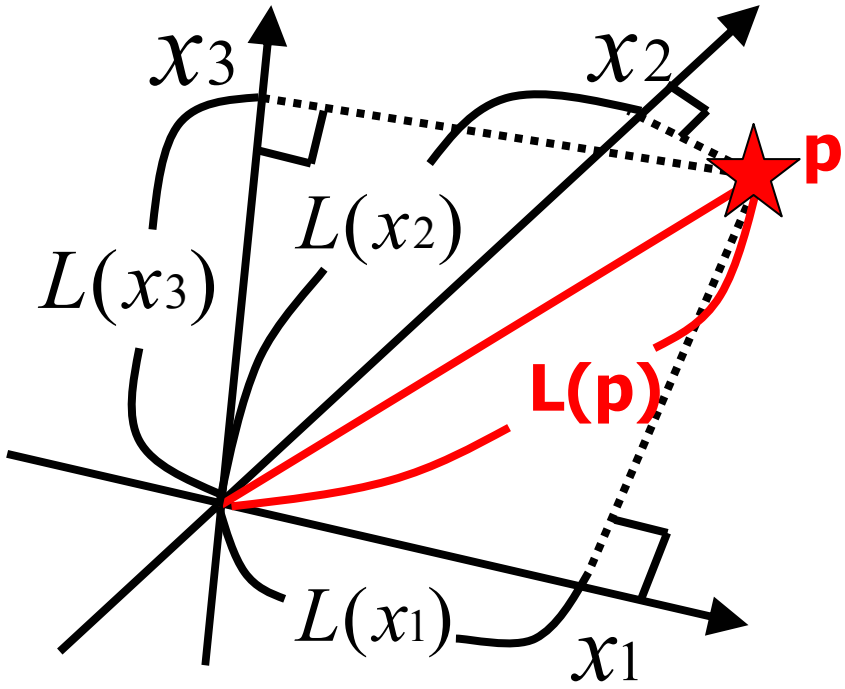
- For n species, no. of splits = 2^{n-1}-n-1 = $O(2^n)$
  - Much smaller than the no. of phylogenetic tree = $O(2^n n!)$
  - Phylogenetic tree likelihood value can be computed easily from its composing split's likelihood values
    - Can reduce likelihood computation from $O(2^n n!)$ to $O(2^n)$

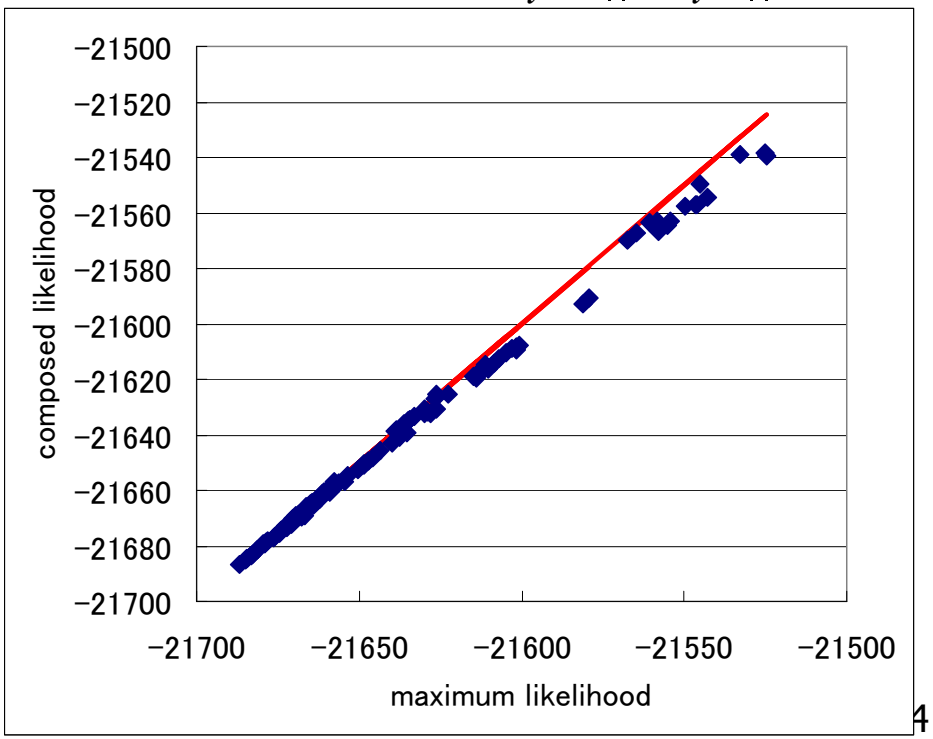
Fewer likelihood computation

Likelihood computation using split decomposition [shimodaira, '01]

- phylogenetic tree P can be represented as composition of m splits
  - Each Split stands for axis of m-dimensional space
  - Likelihood of each split can be considered as the projection of P to each axis
- Approximately compute phylogenetic tree likelihood using likelihood of splits

$$L(p) - L(0) = 1_n' A (A'A)^{-1} v \quad \text{where} \quad A_{\bullet j} = L(x_j) - L(0), \quad v_i = \| A_{i\bullet} \|$$



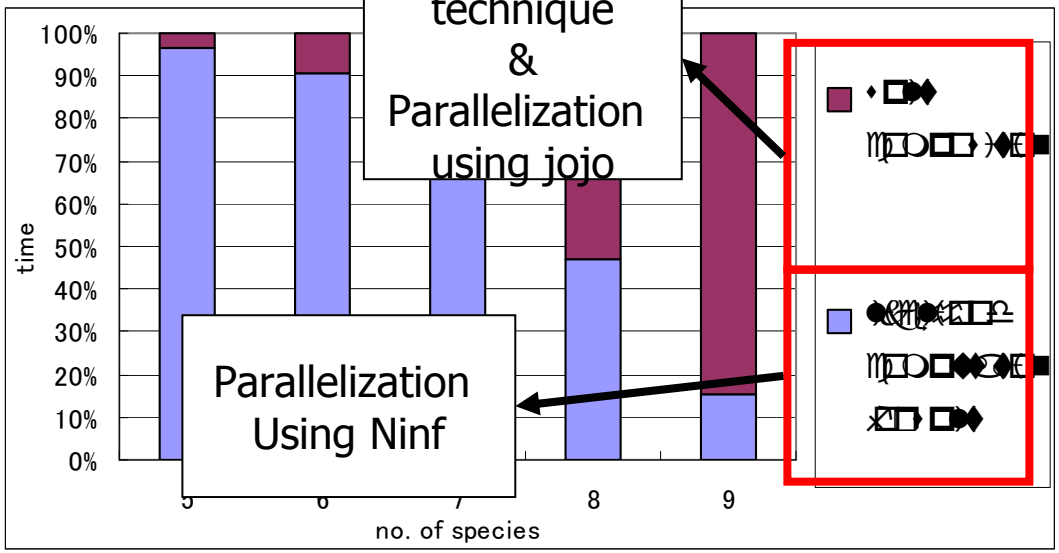


Split composition

- Naïve method : obtain likelihood for each phylogenetic tree one by one
- Split composition : obtain likelihood using split composition
  - = obtain likelihood for each split
  - + Split composition

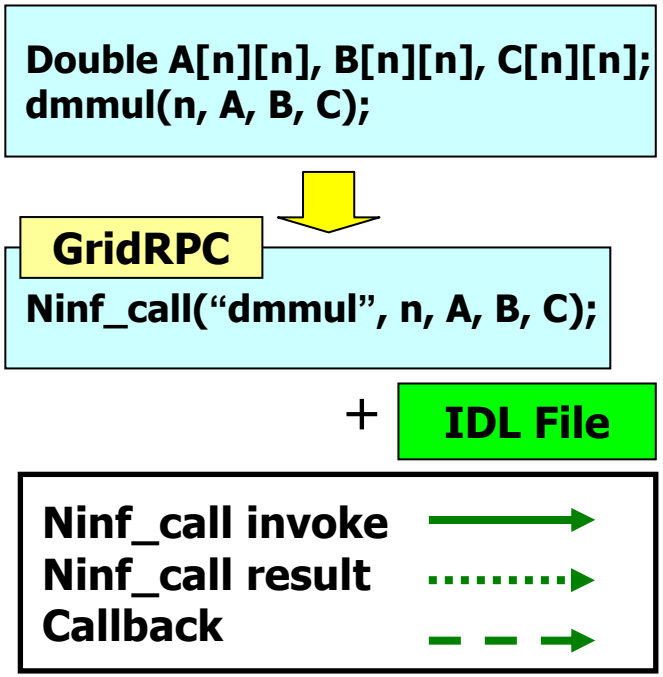
No. species	Naïve method	Splits composition	Speed up
5	9min23sec	3min7sec	3.0x
6	2hour30mn	6min55sec	21.4x
7	1day15hours	35min20sec	66.9x
8	29days	2hour44min	254.6x
9	1year5months	18hours41min	805.8 x

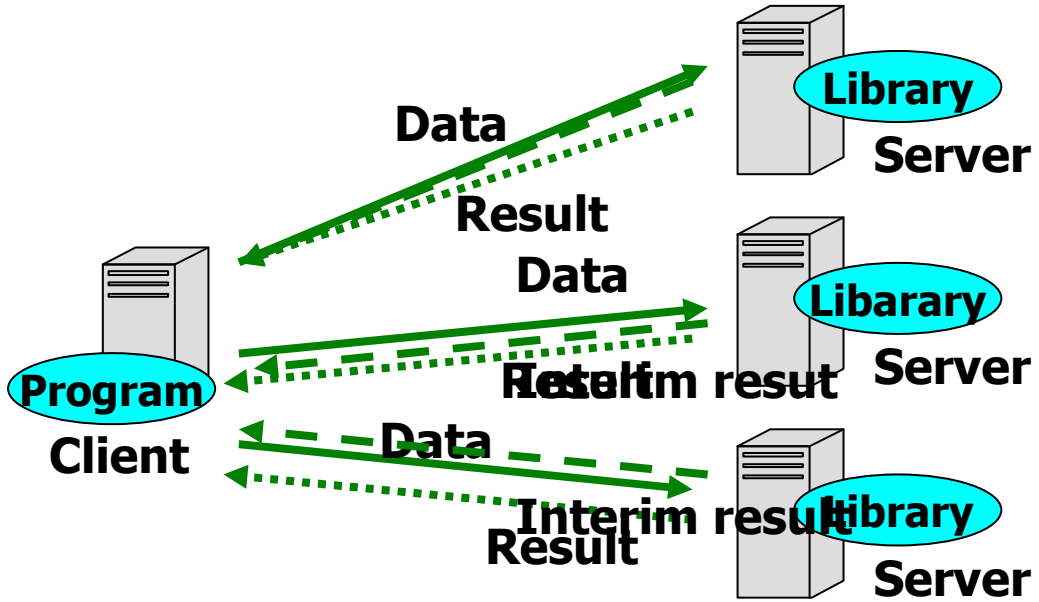
Breakdown of likelih tion using splits



Parallelization with Ninf

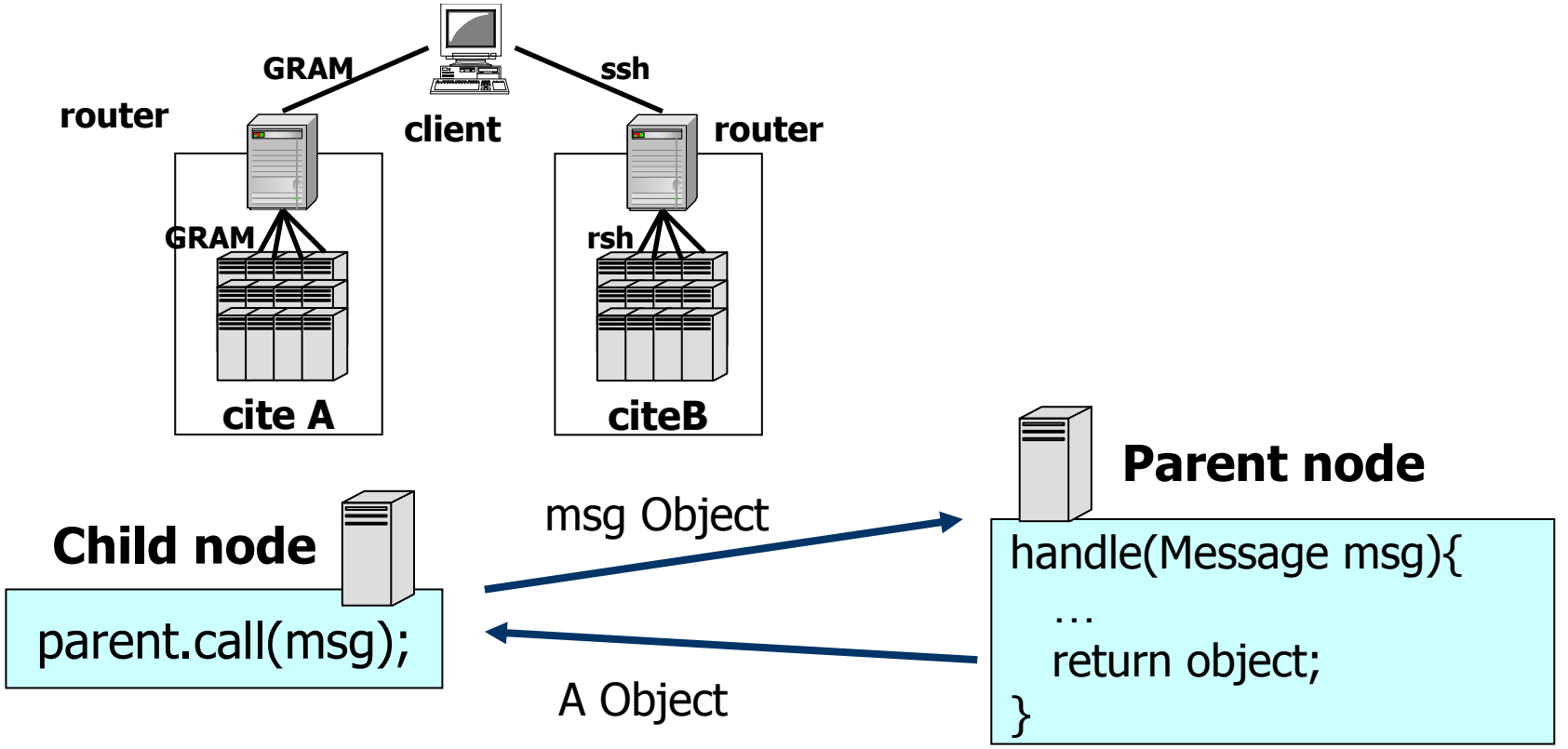
- Ninf [Sekiguchi et al., '96] : A GridRPC system
  - Servers provide computational resource and programs
  - Client invokes a function installed on the server via network
    - The API is designed to minimize the program modification
  - Parallelization with asynchronous invocation
  - The remote function can 'callback' a function in the client





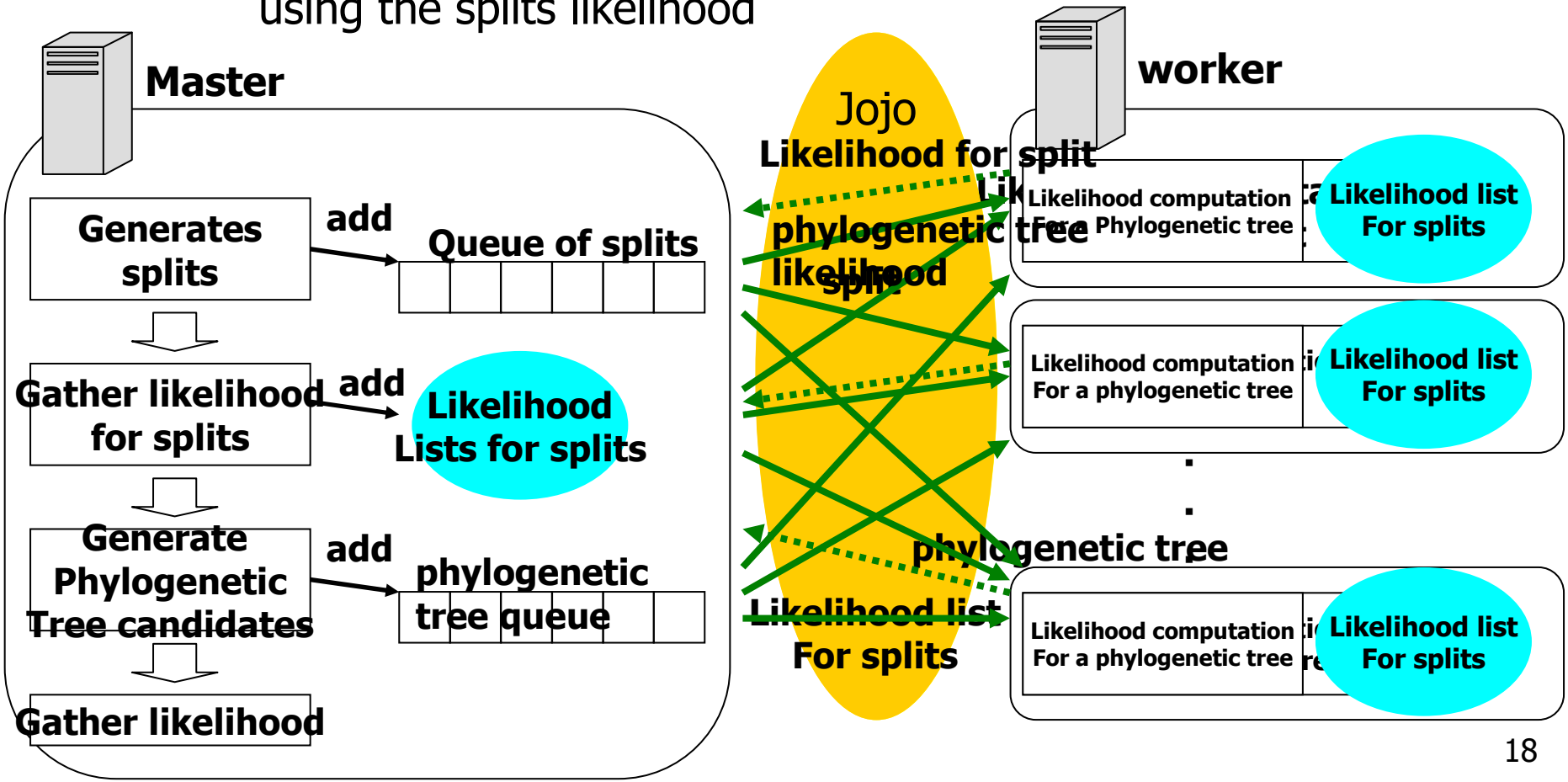
Parallelization with Jojo

- Jojo [nakada et al. 03]: message passing library for Java
  - Dynamic on-demand loading of system/user programs
  - Simple API just allow single object passing



Overview of proposed system

- Use Jojo and Ninf
- Obtain likelihood for every splits
- Generate phylogenetic tree candidates and compute likelihood using the splits likelihood

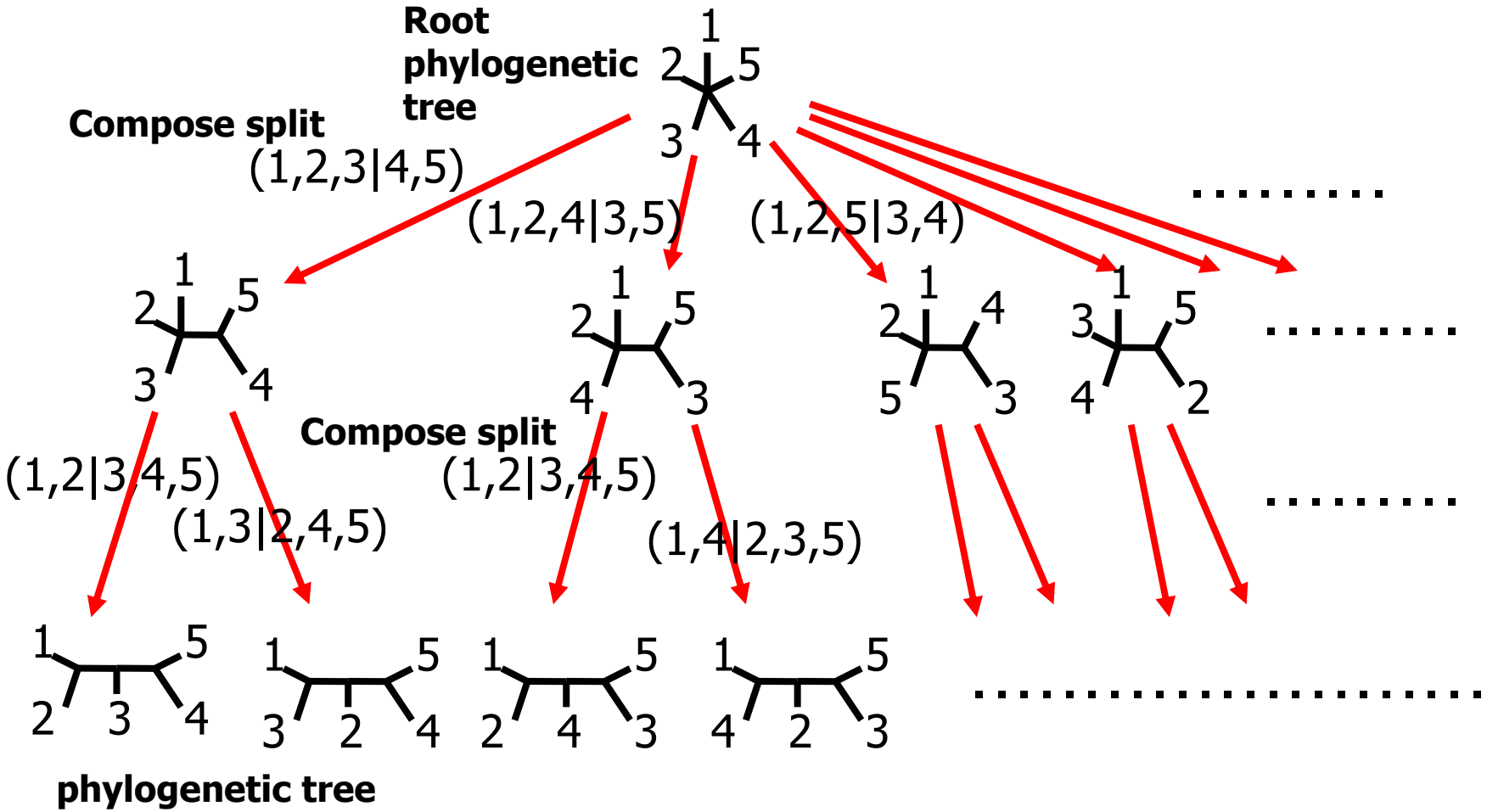


Reduce computation using Combinatorial optimization techniques

- Combinatorial optimization techniques to avoid enumerating every possible phylogenetic tree
  - Branch and bound
    - Cut off useless computation
    - Can obtain optimum
  - Simulated Annealing
    - Approximate method – can gain huge speed up
    - Can be parallelized – Replica exchange method
  - Genetic algorithm
    - Can be parallelized
- In this presentation
  - Branch and bound

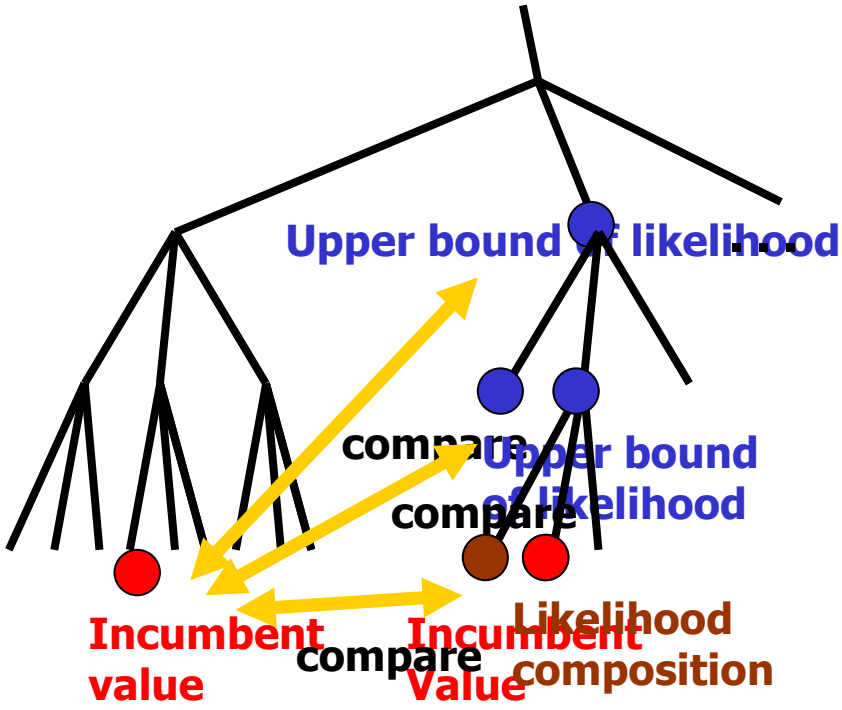
Branch and bound

- Branch and bound search tree
  - To get a valid phylogenetic tree, compose n-3 splits serially, to the Root phylogenetic tree (star-shape)



Branch and bound

- Prune branch on a search tree
  - Maintain incumbent likelihood
  - Compute upper bound for the target node
  - If the upper bound is larger than the incumbent value, proceed computation on the node

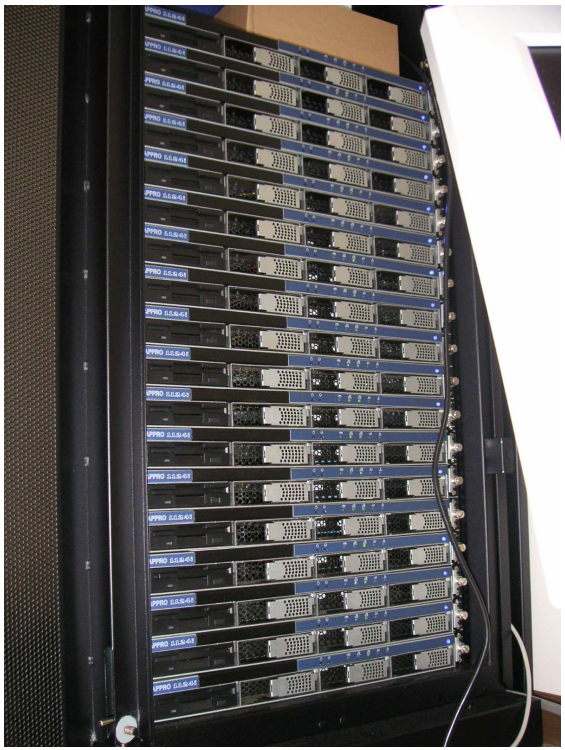


- upper bound for a node
  - Compose all possible splits
  - Take the likelihood as the upper bound

Evaluation

- Experimental environment
  - A cluster called ‘abacus’ installed at TITECH
    - 21 Nodes
    - Linux2.4.18 / GNU Debian woody
    - With 2, 4, 8, 16 workers

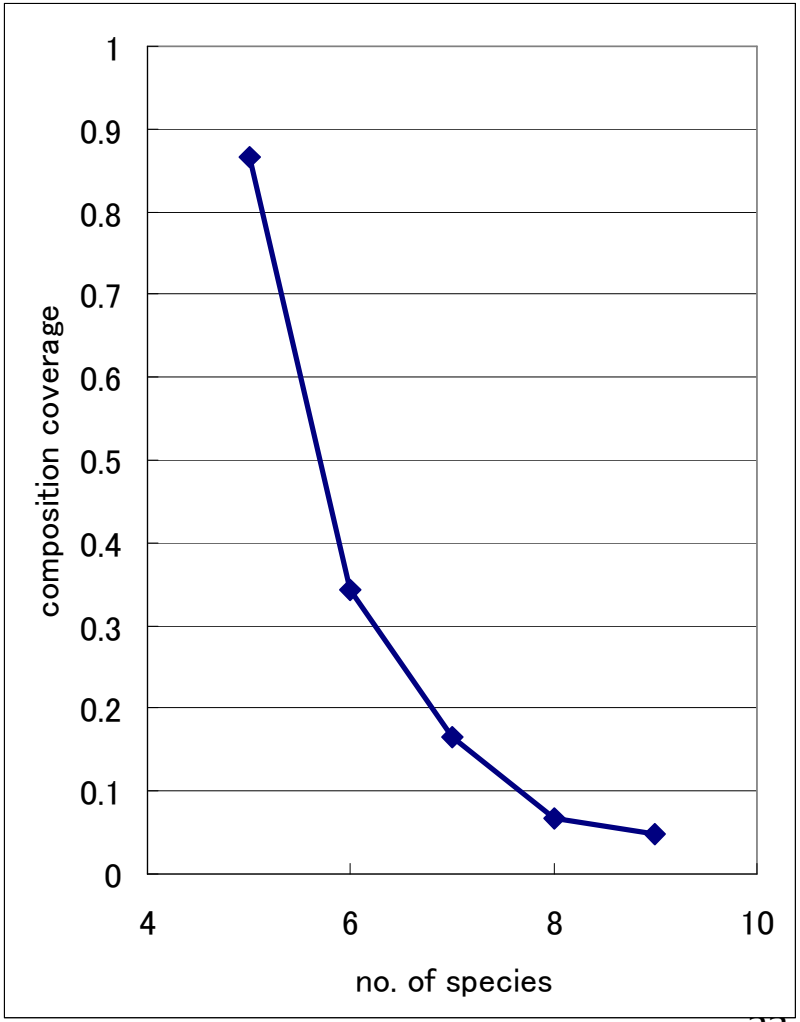
Processor	AMD AthlonMP 2800+ × 2
Memory	1024MB
Network	100BaseT–Ethernet



- Sample group
  - seal, cow, rabbit, opossum, mouse, Human, dugong, armadillo, rat
  - Use mitochondria Sequence – downloaded from NCBI
    - 3392 amino acids
- Likelihood computation
  - Codeml of Paml

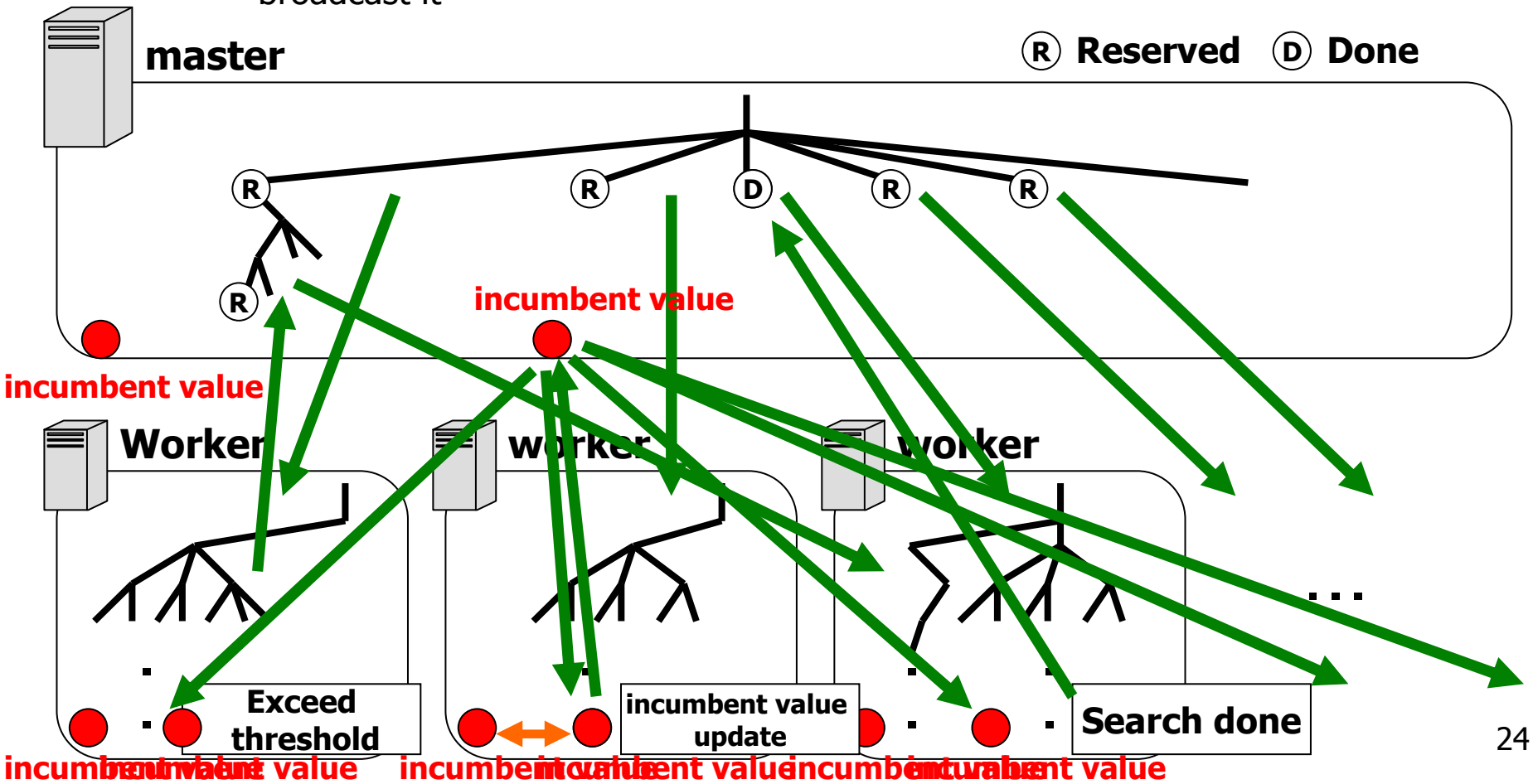
Result of Branch and bound

- Search method
  - Order splits by likelihood
  - Depth first
- Composition coverage
  - Whole composition times = leaf node ＋prune
  - Composition coverage = whole composition times / whole no. of phylogenetic tree
- Cut 95.3 % of trees for 9 species
  - 19.9 times faster



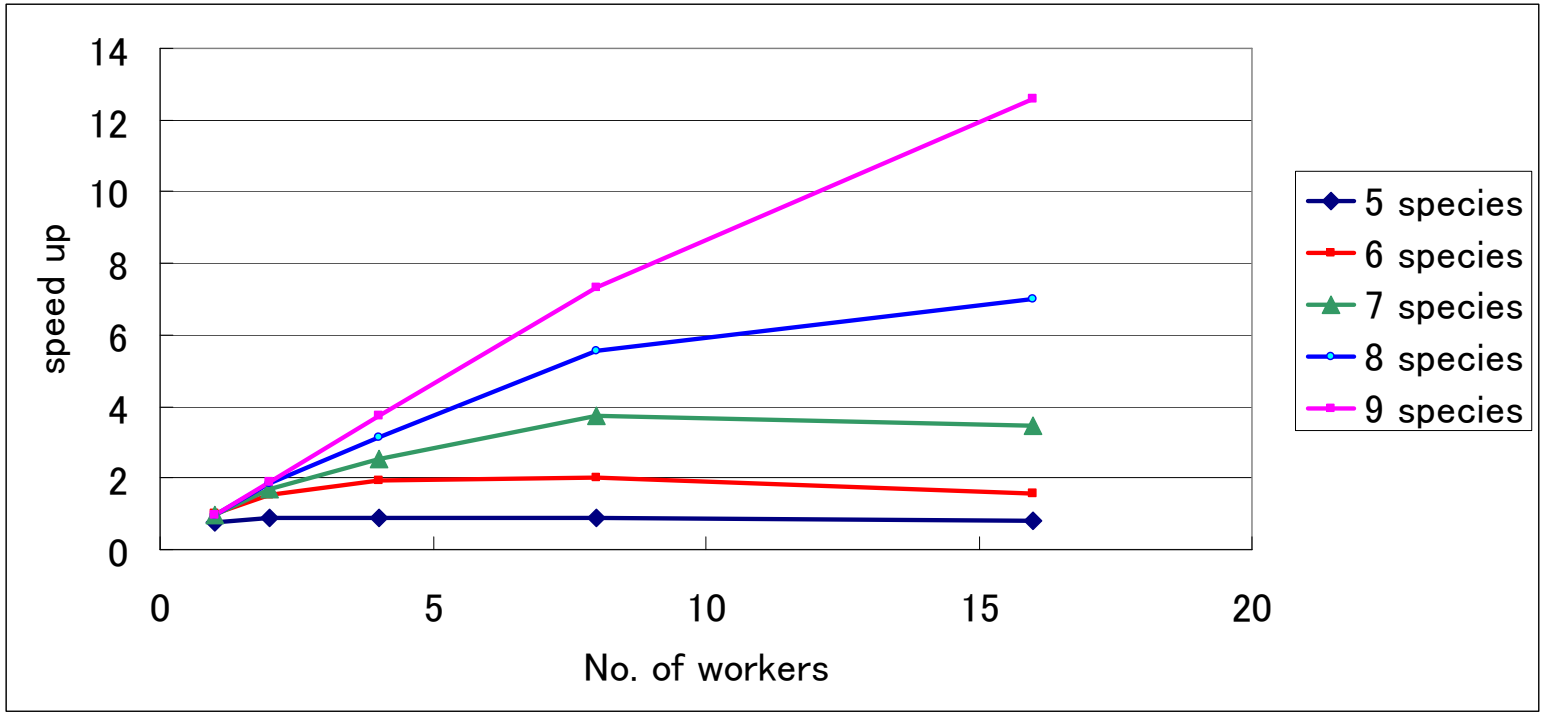
Parallelization of branch and bound with Jojo

- Assign a subtree for each worker
- Worker returns sub problems that exceed a specified threshold
- Worker requires new subtree for the master
- Worker reports newly found incumbent value for the master, and master broadcast it



Result of Parallel Branch and bound

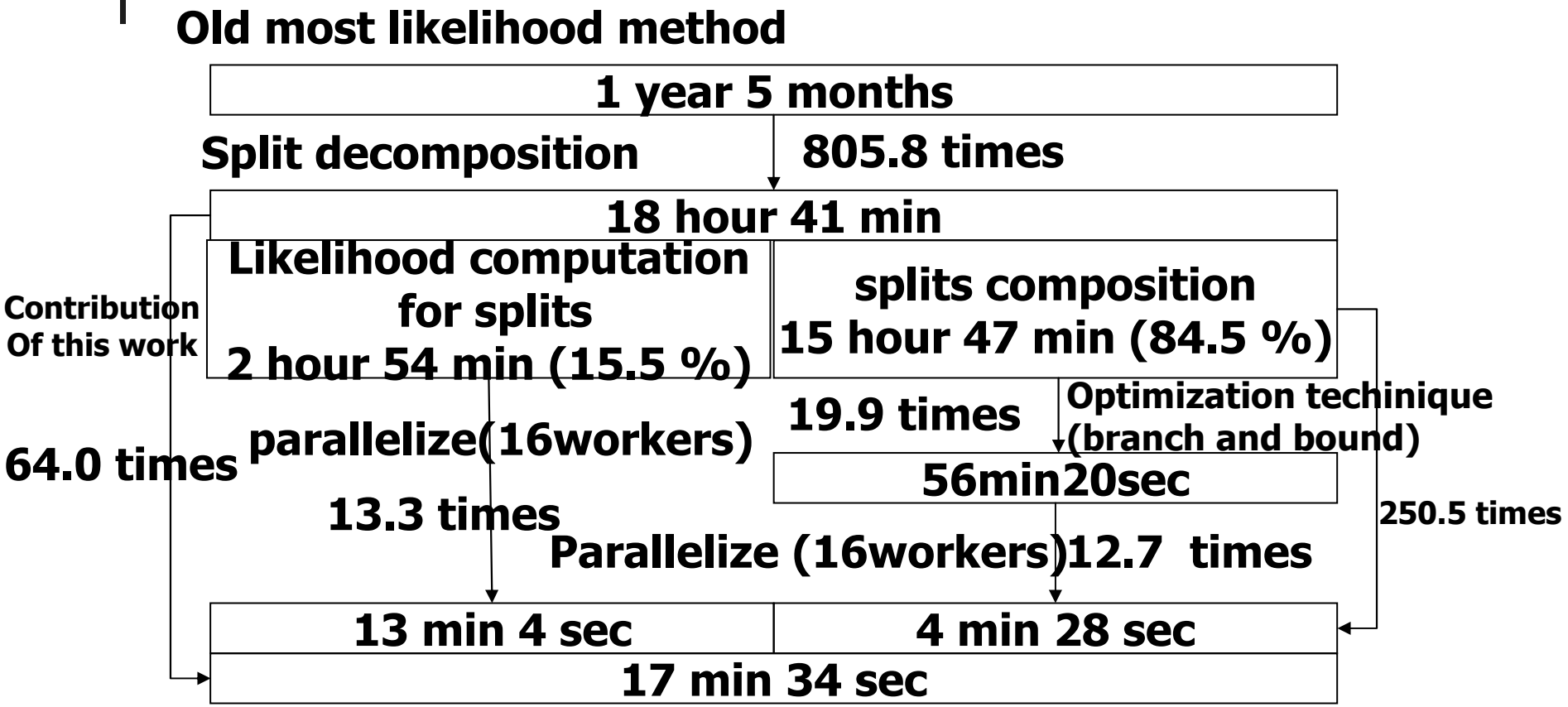
- Speed up
  - No. of workers 2, 4, 8, 16,
  - No. of species 5, 6, 7, 8, 9
  - 12.8 time faster, for 16 workers , with 9 species
    - 80% efficiency



Speed up for parallelized branch and bound

Summary

- Speed up for 9 species

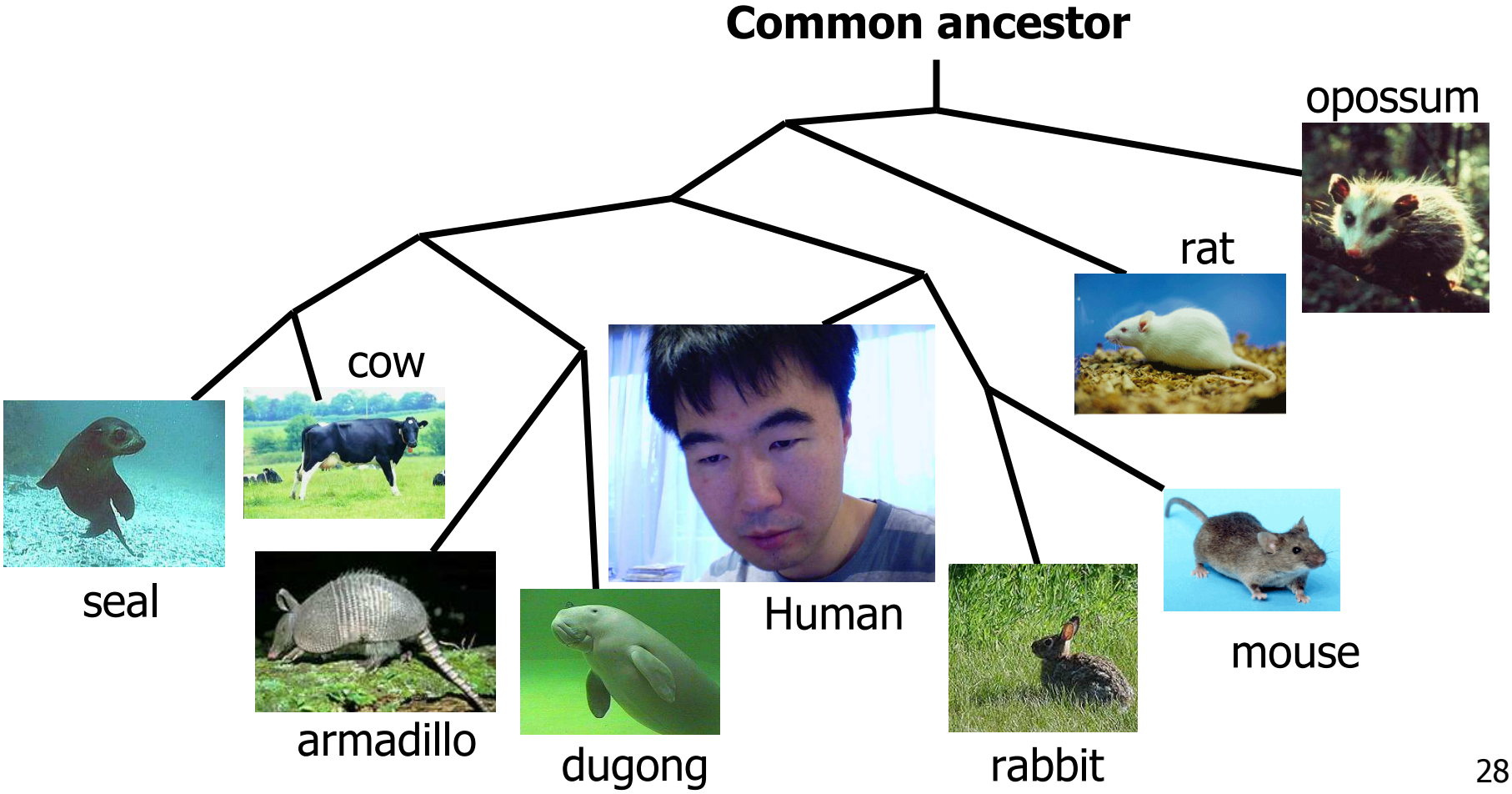


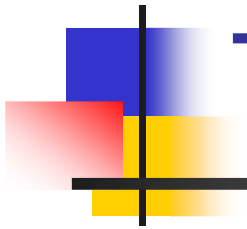
51493.3 times

Future work

- Employ Genetic Algorithm for combinatorial optimization
- Use other likelihood programs
  - Use other sequence such as protein
- Scalability Evaluation
  - Evaluation on Grids
- Fault Tolerance

The result

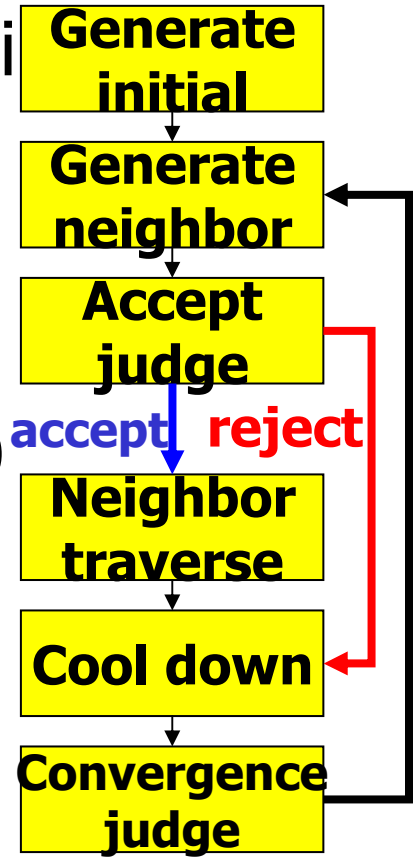

Common ancestor
opossum
rat
cow
seal
armadillo
dugong
Human
rabbit
mouse



Thank you

Combinatorial optimization technique2：Simulated annealing

- Generate ‘neighbor’ and accept or reject it depending on the E
- Rarely fall into local maximum poi
  - Cooling scheduling is important
  - $T_{next} = \alpha T_{current}$
    - α :cooling parameter ($0 << \alpha < 1$)



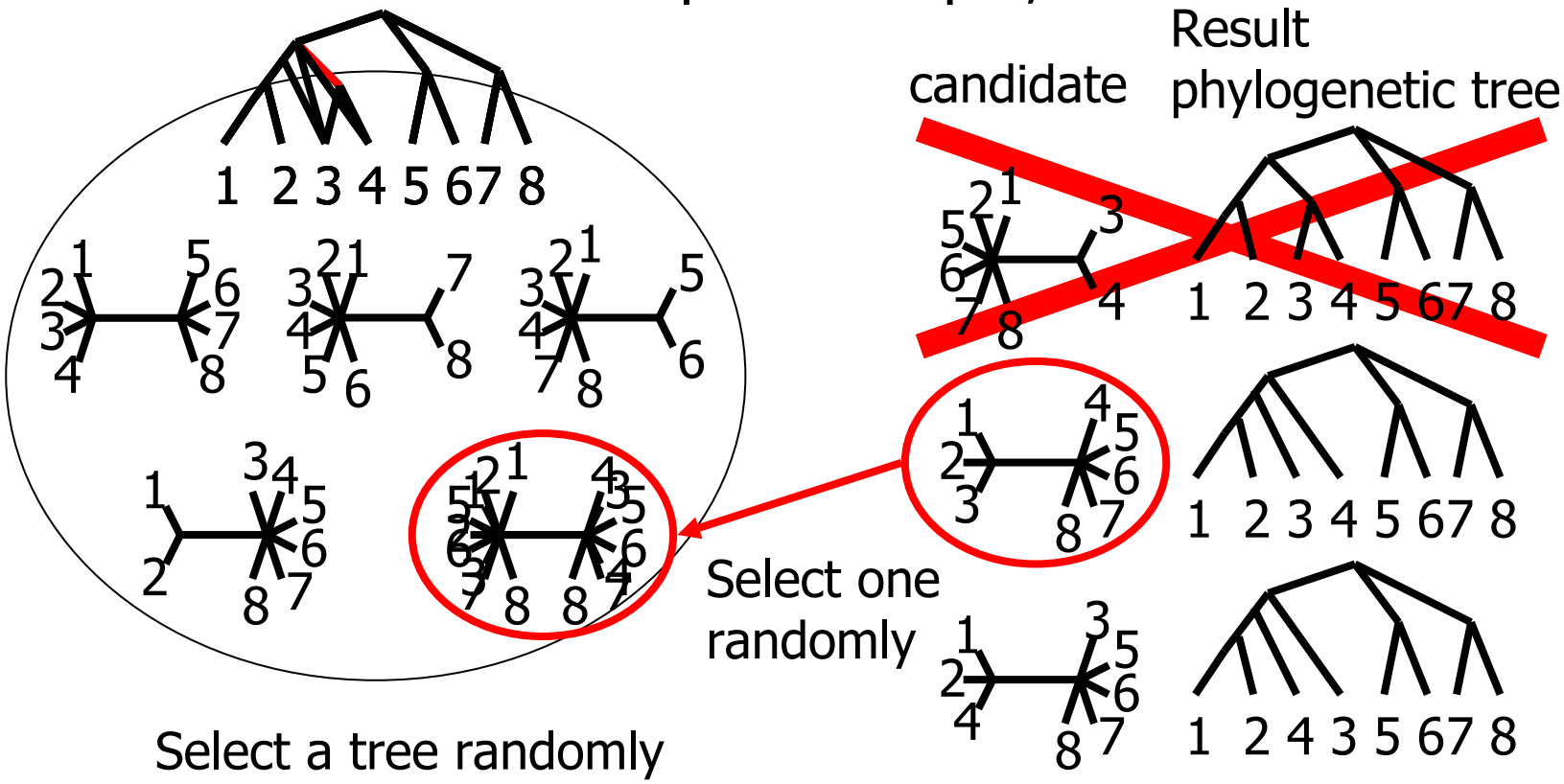
Accept judgement ($\Delta E = E_{next} - E_{current}$)

$$\begin{cases} \Delta E < 0 \Rightarrow \textbf{accept} \\ \Delta E > 0 \Rightarrow \textbf{Accept if} \end{cases}$$

$$\exp\left\{-\frac{\Delta E}{T}\right\}$$

“Neighbor” for annealing

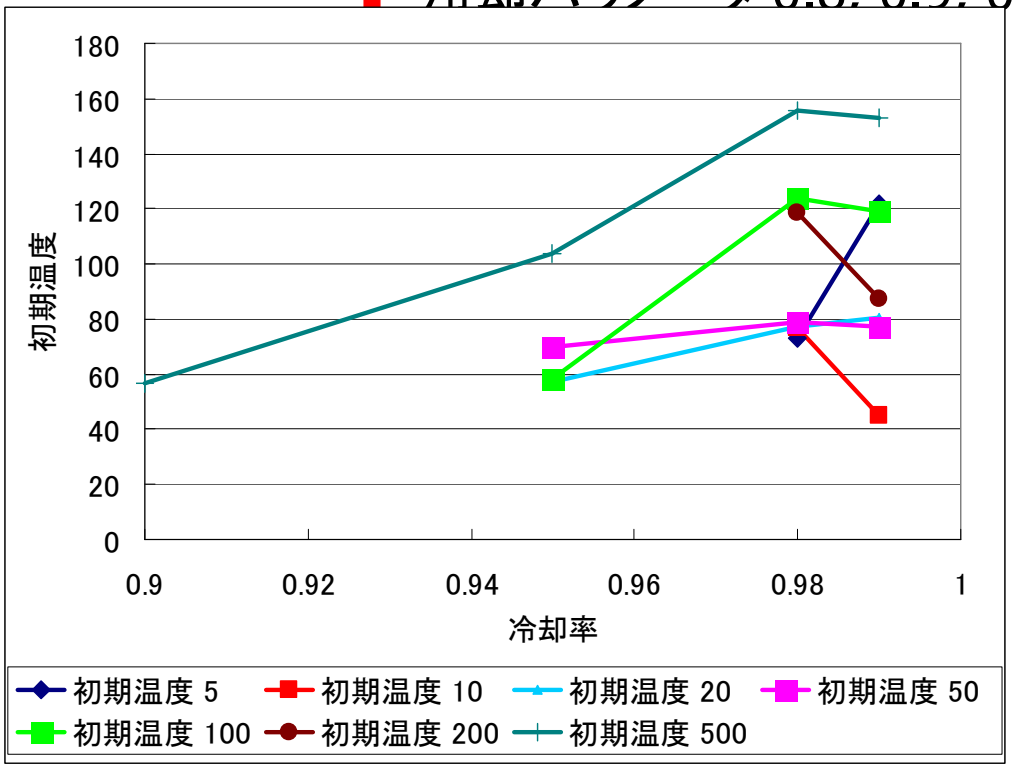
- For Simulated annealing, we have to define neighbor
- Remove a split from the target tree,
- And add another possible split,



焼きなまし法のパラメータと結果

- 初期温度と冷却法について評価
  - 探索回数で評価
    - 上位 1 % のphylogenetic treeに 10 回推移することを終了条件とした
  - 初期温度 5, 10, 20, 50, 100, 200, 500
  - 冷却パラメータ 0.8, 0.9, 0.95, 0.98, 0.99

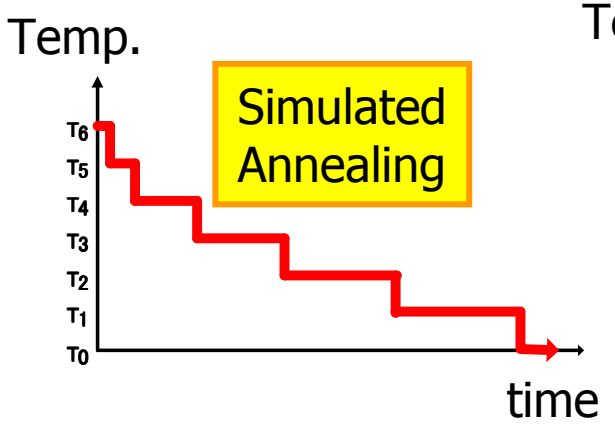
生物種 7 種で評価

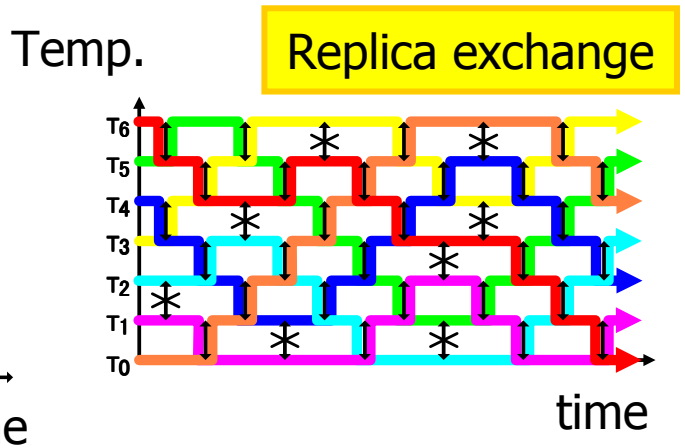


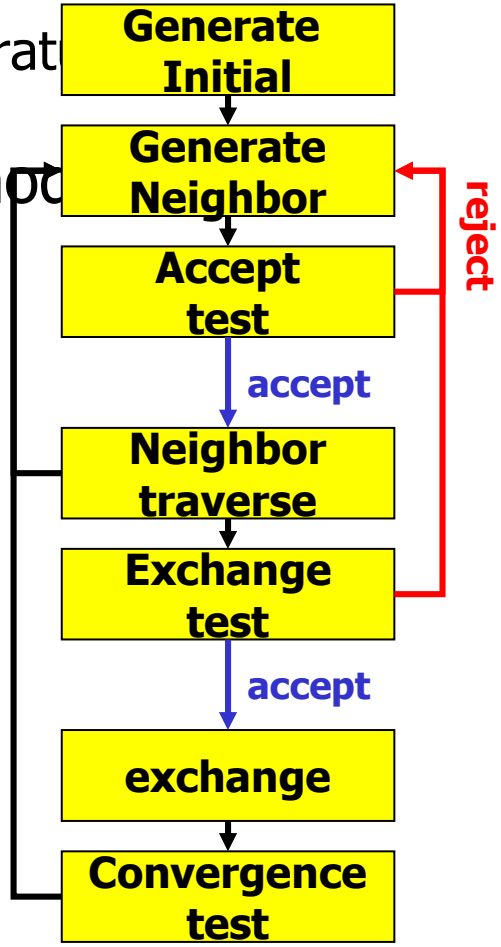
- 初期温度が低いほうが収束まで速い
- 冷却パラメータ が低いと局所解に収束
- 初期温度 10 冷却パラメータ 0.99
  - 収束率 100%
  - 平均探索回数 45 回
  - 探索回数 95.2% 削減

Parallelize Simulated annealing

- Replica exchange method
  - Have several 'replica's and assign temperat
    - Periodically exchange the temperature
- Advantage of Replica exchange method
  - No temp. scheduling required
  - May speed up even for one CPU
  - Easy to parallelize



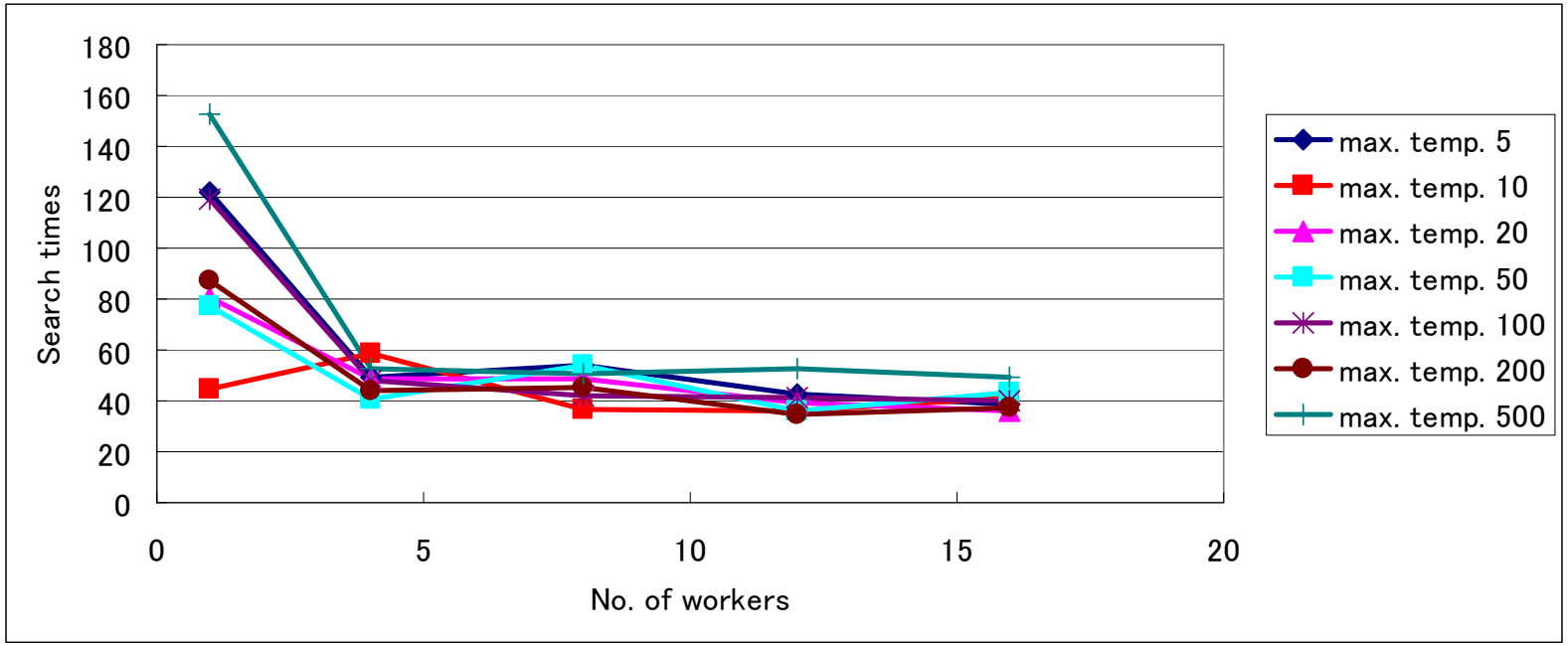




Result of Replica exchange method

- Measure: minimum search 全ワーカの最小探索回数
  - 最大温度に関わらず安定的な探索回数の削減
    - 合成回数が 60 回以下（削減率 92.9 %以上）
  - 収束性の向上
    - 4 ノード以上の全実験で収束
  - No speed up gained for larger no. of workers
    - より大きな問題で評価する必要がある

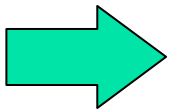
For 7 species



レプリカ交換法での探索回数

Background

- Phylogenetic Tree Inference based on DNA / protein sequences
  - Determine probability for each tree using statistics
  - Requires enormous computation
    - Huge number of Phylogenetic tree
    - Likelihood computation for each tree requires certain amount of computation
- Parallel computing
  - Clusters、Grid
  - Good Cost-performance



Speeding up Phylogenetic tree inference by parallelization

Related work

- RAxML [Alexandros et al., ‘03]
  - Depends largely on the initial value
  - Master-worker with MPI
- MrBayes 3 [Ronquist et al., ‘03]
  - The new generation of MrBayes [Ronquist et al., ’01]
  - Provides framework for phylogenetic tree inference
  - MPI implementation

- Contribution of this work
  - Efficient computation using splits
  - Parallelization aiming to the Grid

Contribution of this work

- Parallelize likelihood approximation
  - Speed up the approximation
    - Try every phylogenetic tree
    - Reduce computation cost for each tree using split composition
  - Parallelization
    - Master-worker style implementation using Ninf: a GridRPC system
- Reduce the no. of phylogenetic trees to be actually tested, using combinatorial optimization technique
  - Employ Branch and Bound method
  - Parallelization of branch and bound
    - Master-worker implementation using Jojo: a Message passing library for Java