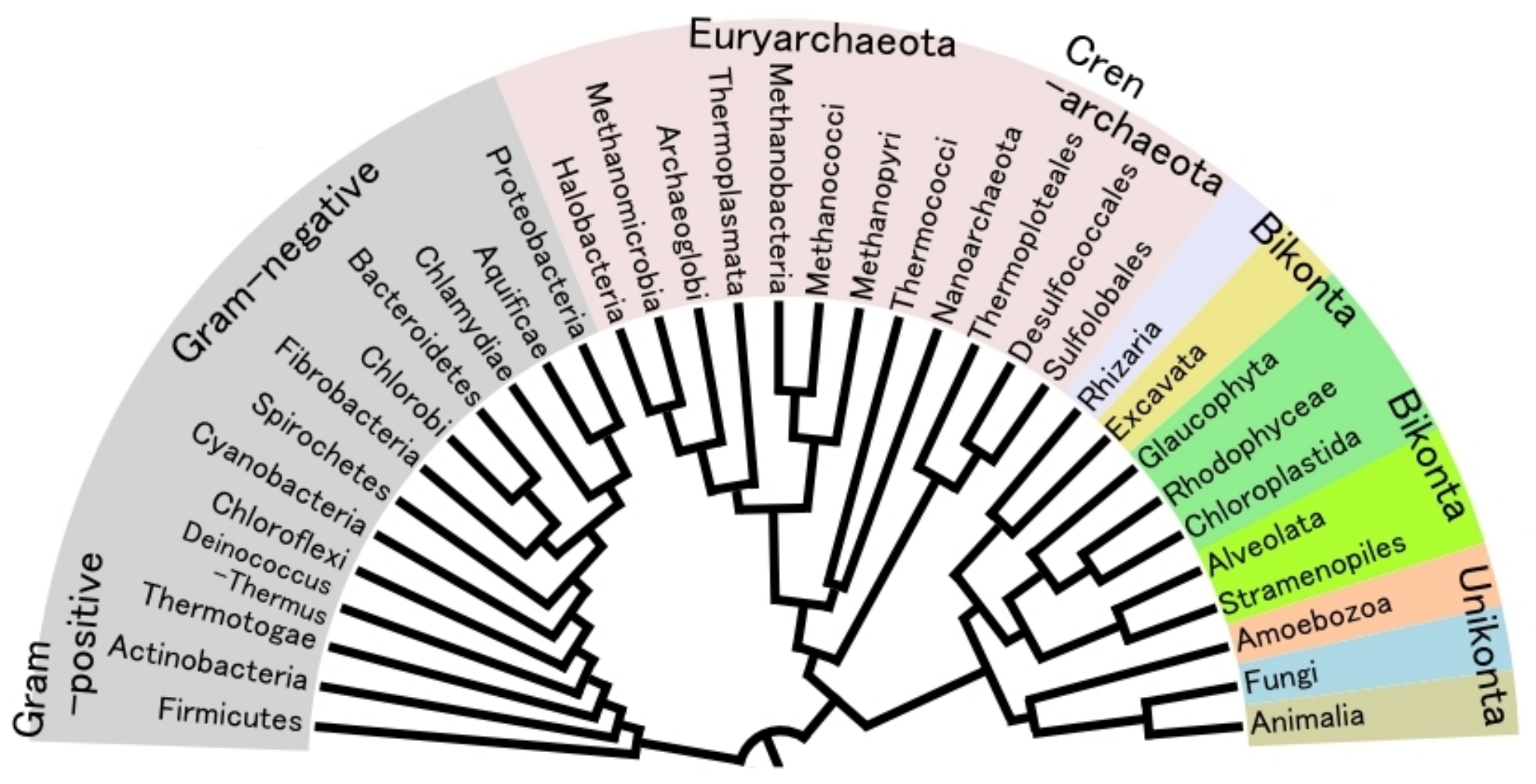
MERGING PARTIALLY LABELLED TREES

Anthony Labarre and **Sicco Verwer**
{ ANTHONY.LABARRE, SICCO.VERWER }@CS.KULEUVEN.BE

KATHOLIEKE UNIVERSITEIT LEUVEN

The BeNeLux Bioinformatics Conference (BBC) 2011

CONTEXT AND MOTIVATION



- Evolution is usually depicted by phylogenetic trees; however:
  1. evolution is not always tree-like (hybridization, horizontal gene transfer, ...)
  2. there may be many equally good trees;
- Phylogenetic *networks* [2] are more appropriate in these cases;
- We focus here on the *minimum common supergraph* approach, initiated by Cassens et al. [1] and formalised by Labarre [3];

PROBLEM

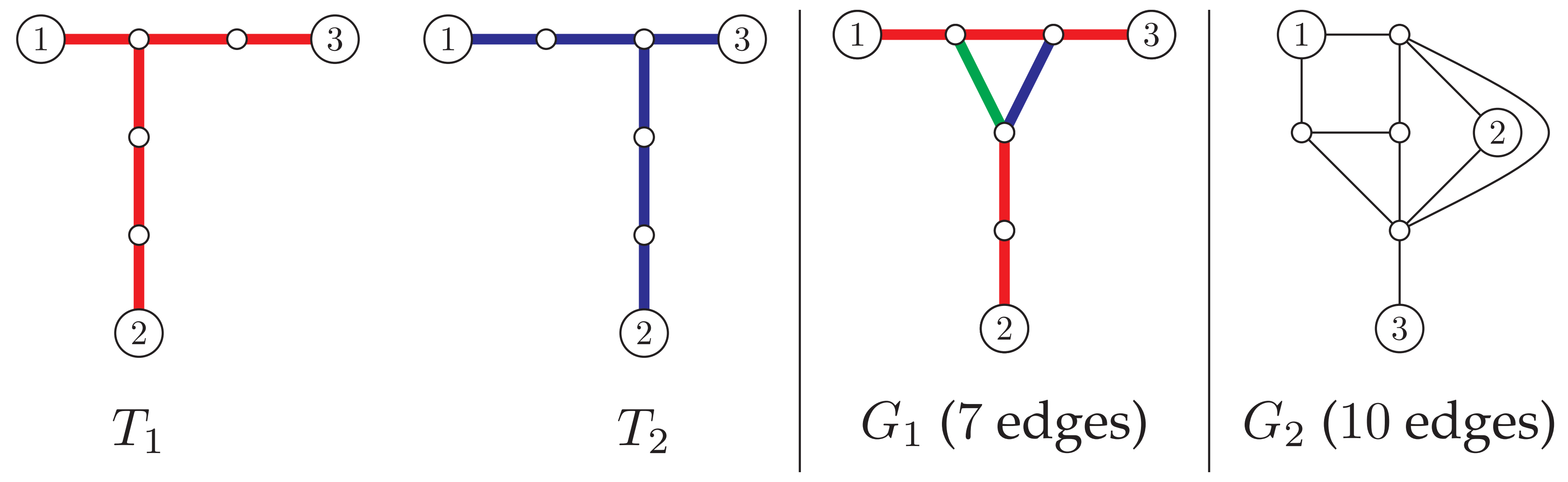
Given: trees $T_1, T_2, \ldots, T_t$.
Find: a graph G which:

1. contains $T_1, T_2, \ldots, T_t$, and
2. has as few edges as possible.

All trees and G have n vertices, k of which are labelled using $\{1, 2, \ldots, k\}$. Labels are used exactly once in each tree and in G.

EXAMPLE

input:



T_1 T_2 G_1 (7 edges) G_2 (10 edges)

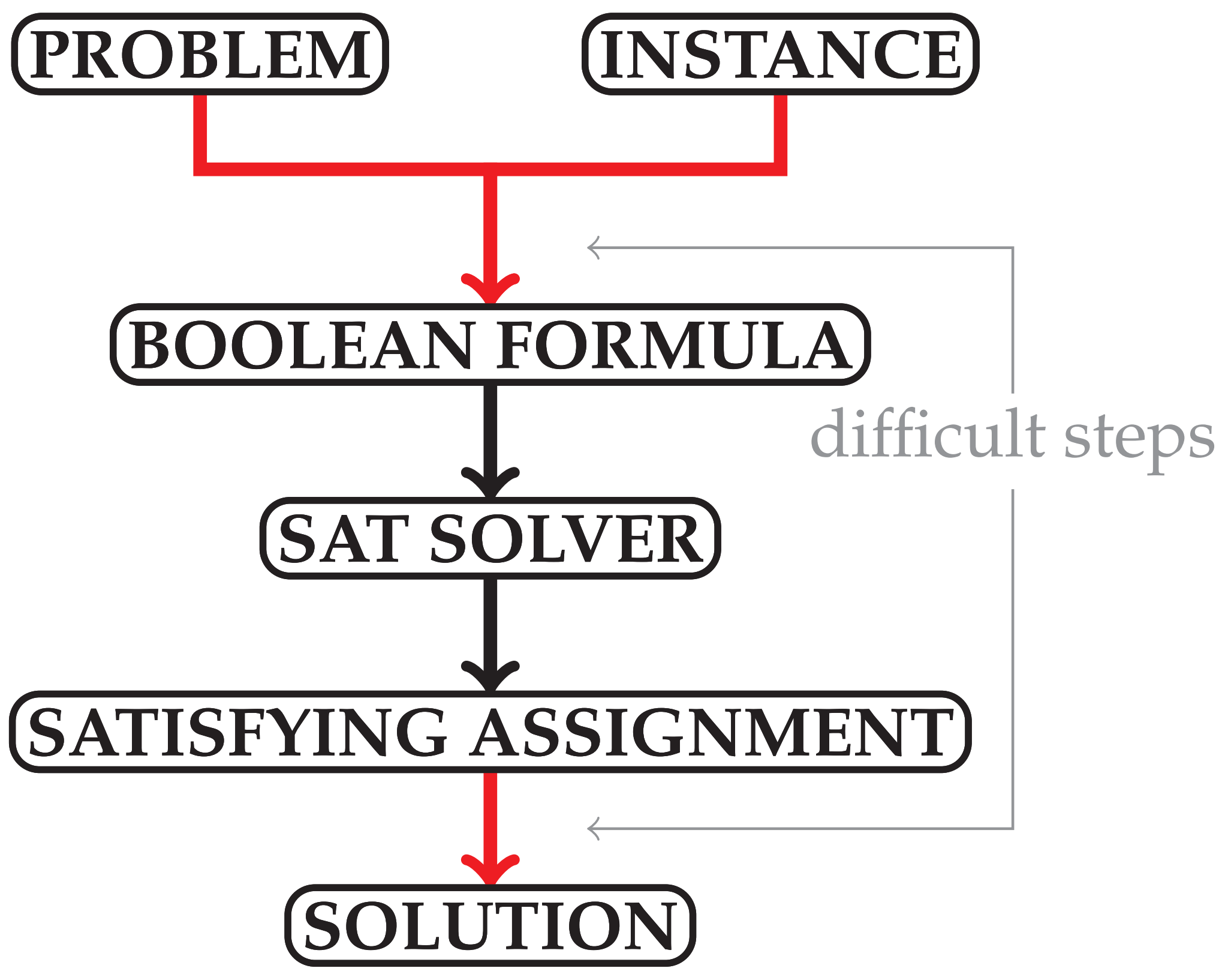
RESULTS

(to appear [4])

1. The problem is NP-hard...
2. ...but it can be solved efficiently in practice;

MORE DETAILS

We use a *SAT solver*; traditionally, this works as follows:



PROBLEM
INSTANCE
BOOLEAN FORMULA
difficult steps
SAT SOLVER
SATISFYING ASSIGNMENT
SOLUTION

IDP [5] allows us to bypass the difficult steps:

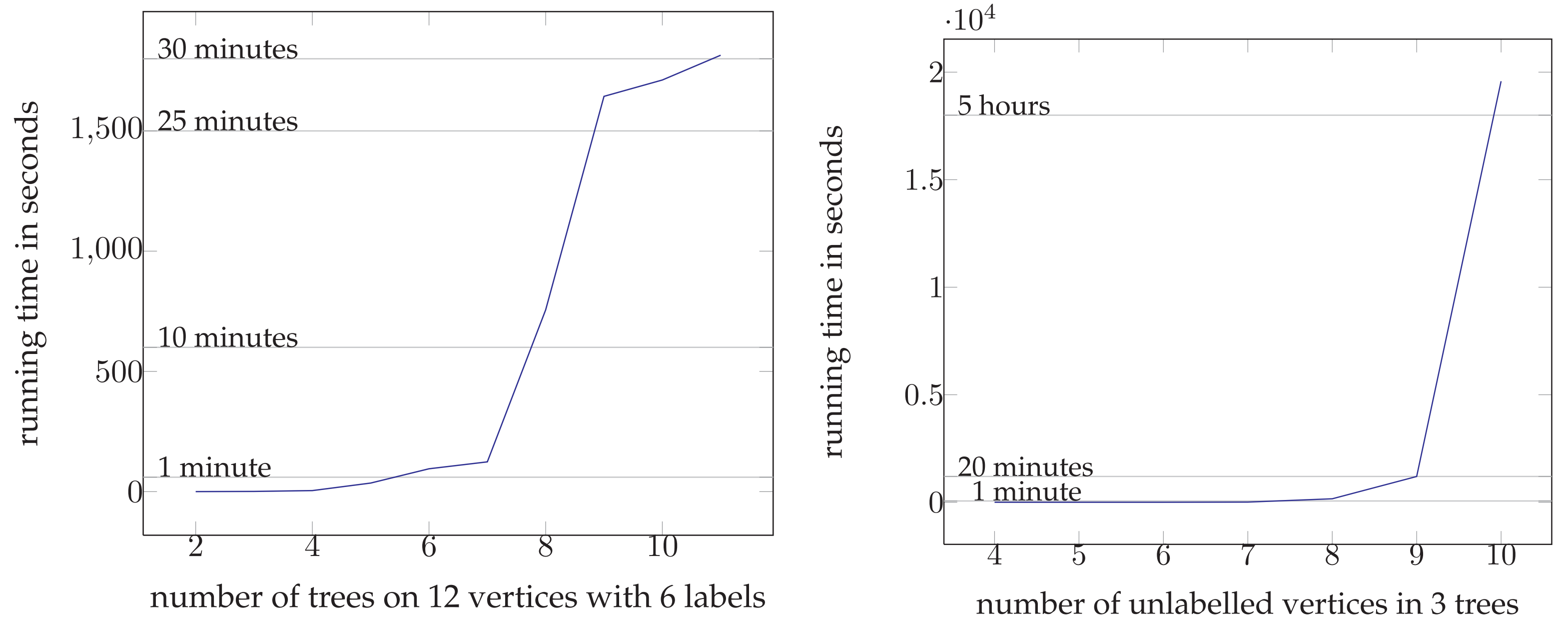
- high-level descriptions of problem and instance;
- solution also returned in a high-level description;

ADVANTAGES

- ✓ Ease of implementation;
- ✓ You can terminate the program at any time and retrieve the current solution;

FINDING AN OPTIMAL SOLUTION

Growth of the running time for an optimal solution (averages over 20 runs):



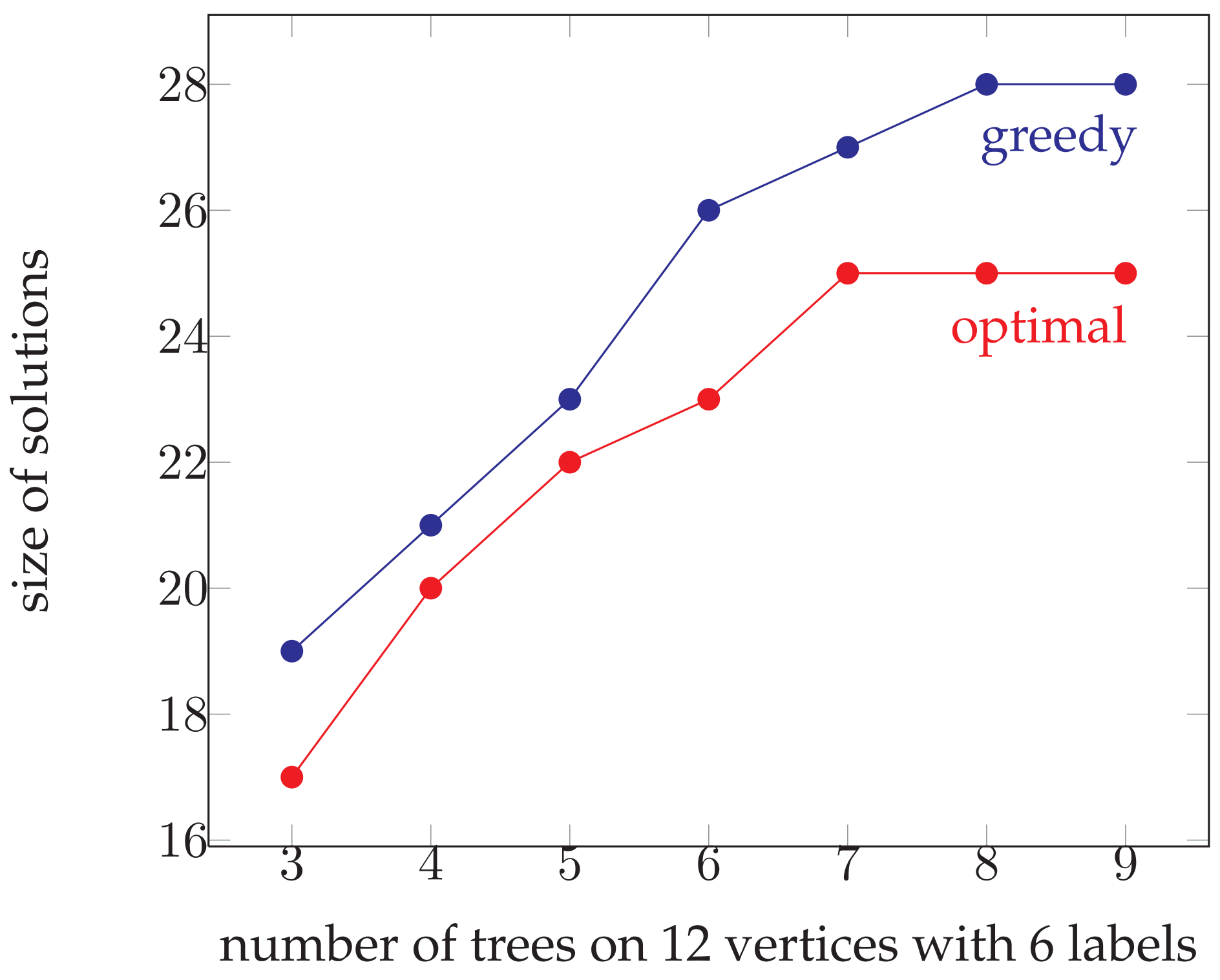
note that the search space is huge: $O((n-k)!^{t-1})$

(Experiments carried out on randomly generated data, on a desktop machine equipped with an Intel(R) Core^{TM} i7 CPU 870 @ 2.93GHz CPU (64bits) with 8GB of RAM)

GREED PAYS OFF

- If dataset is too large, a greedy approach is much faster and performs very well:
  1. compute the size of an optimal solution for each pair of trees;
  2. merge the two "closest" trees (w.r.t. solution size);
  3. keep merging the resulting supergraph with the closest tree;
  4. stop when all trees have been merged.
- Here's the kind of quality one can expect:

no timeout:



greedy is here at most 13% "worse" than the optimal solution

timeout=2000 ms, averages over 4 runs:

			solution sizes	
#trees	#nodes	#labels	exact	greedy
5	10	5	17.50	18.00
10	10	5	19.50	21.50
20	10	5	23.00	25.25
5	20	5	34.75	32.50
5	20	10	53.00	46.00
10	20	5	38.75	35.25
10	20	10	64.25	56.50
20	20	5	42.25	42.25
20	20	10	75.50	71.75
5	50	5	130.00	131.25
5	50	10	128.00	132.75
5	50	25	207.75	184.75
10	50	5	183.75	154.50
10	50	10	177.75	154.75
10	50	25	270.00	269.25
20	50	5	241.50	171.75
20	50	10	232.00	152.25
20	50	25	346.25	279.00

cases where greedy "wins"

REFERENCES

[1] I. CASSENS, P. MARDULYN, AND M. C. MILINKOVITCH, *Evaluating intraspecific "network" construction methods using simulated sequence data: Do existing algorithms outperform the global maximum parsimony approach?*, Systematic Biology, 54 (2005), pp. 363–372.

[2] D. H. HUSON, R. RUPP, AND C. SCORNAVACCA, *Phylogenetic Networks: Concepts, Algorithms and Applications*, Cambridge University Press, Dec. 2010.

[3] A. LABARRE, *Combinatorial aspects of genome rearrangements and haplotype networks*, PhD thesis, Université Libre de Bruxelles, Brussels, Belgium, Sept. 2008.

[4] A. LABARRE AND S. VERWER, *Merging partially labelled trees: hardness and an efficient practical solution*, (2011). In preparation.

[5] J. WITTOCX, M. MARIËN, AND M. DENECKER, *The IDP system: a model expansion system for an extension of classical logic*, in Proceedings of the Second International Workshop on Logic and Search, Computation of Structures from Declarative Descriptions (LaSh), Leuven, Belgium, Nov. 2008, pp. 153–165.