

Integer Programming for NP- hard Phylogenetic (and Population- Genetic) Problems

D. Gusfield

Isaac Newton Institute

September 4, 2007

Phylogeny problems often have data with

- > Missing entries
- > Homoplasy
- > Genotype (conflated) sequences, rather than simpler haplotype sequences

Most of these problems are NP-hard, although some elegant poly-time solutions exist (and are well-known) for special cases.

Question

Can Integer Programming efficiently solve these problems in practice on useful ranges of data? Clearly not genomic or tree-of-life scale.

We have recently developed ILPs for over fifteen such problems and intensively studied their performance (speed, size and biological utility). We discuss five related problems in this talk.

Starting Model: Compatibility, Perfect Phylogeny, with binary sequences

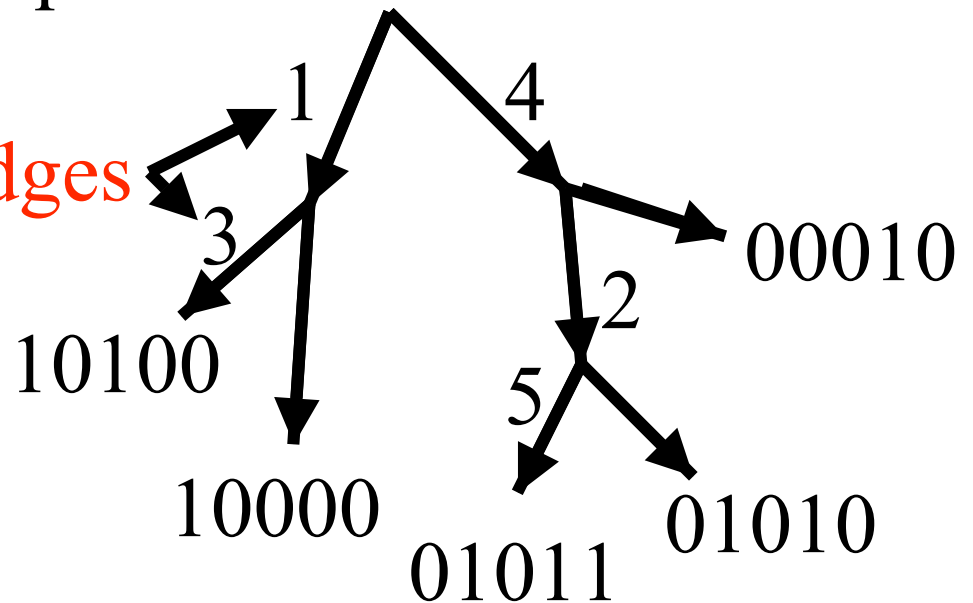
Only one mutation per site
allowed.

sites 12345
Ancestral sequence 00000

Site mutations on edges

The tree derives the set M:

10100
10000
01011
01010
00010



Extant sequences at the leaves

Everyone here knows

Classic NASC: A set of sequences with **no** duplicate columns can be generated on a **unique** perfect phylogeny if and only if no two columns (sites) contain all four binary pairs (gametes):

0,0 and 0,1 and 1,0 and 1,1

This is the 4-Gamete or Compatibility Test

And

The set of splits of a tree uniquely determine the tree.

A pair of sites that has all four binary pairs is called **incompatible**, otherwise is called **compatible**.

For M of dimension n by m , the existence of a perfect phylogeny, or the test for pairwise compatibility for M , can be tested in $O(nm)$ time and a tree built in that time, if there is one.

Problem M1: Perfect Phylogeny with Missing Data

Given **binary** sequences M with some ? entries, change each ? to 0 or 1 in order to **minimize** the resulting number of incompatible pairs of sites.

Special Case (Existence Problem):

Determine if the ?s can be set to 0, 1 so that there are **no** resulting incompatibilities.

NP-hard in general, but if the root of the required tree is specified, then the problem has an elegant poly-time solution (Pe'er, Sharan, Shamir).

Simple ILP for Problem M1

If cell (i,p) in M has a $?$, create a binary variable $Y(i,p)$ indicating whether the value will be set to 0 or to 1.

For each pair of sites p, q that **could be made** incompatible, let $D(p,q)$ be the set of missing or **deficient** gametes in site pair p,q , needed to make sites p,q incompatible.

For each gamete a,b in $D(p,q)$, create the binary variable $B(p,q,a,b)$,
and create inequalities to set $B(p,q,a,b)$ to 1 **if** the Y variables for cells for sites p,q are set so that gamete a,b is created in **some** row for sites p,q .

Example

p q	
0 0	$D(p,q) = \{1,1; 0,1\}$
? 1	
1 0	
? ?	
? 0	
0 ?	

To set the B variables, the ILP will have inequalities for each a,b in D(p,q), one for each row where a,b could be created in site pair p,q.

For example, for a,b = 1,1 the ILP has:

$$Y(2,p) \leq B(p,q,1,1) \quad \text{for row 2}$$

$$Y(4,p) + Y(4,q) - B(p,q,1,1) \leq 1 \quad \text{for row 4}$$

Example continued

p q

0 0
? 1
1 0
? ?
? 0
0 ?

For $a, b = 0, 1$ the ILP has:

$$Y(2,p) + B(p,q,0,1) \Rightarrow 1 \quad \text{for row 2}$$

$$Y(4,q) - Y(4,p) - B(p,q,0,1) \leq 0 \quad \text{for row 4}$$

$$Y(6,q) - B(p,q,0,1) \leq 0 \quad \text{for row 6}$$

The ILP also has a variable $C(p,q)$ which is set to 1 if **every** gamete in $D(p,q)$ is created at site-pair p,q .

In the example:

$$B(p, q, 1, 1) + B(p, q, 0, 1) - C(p,q) \leq 1$$

So, $C(p,q)$ is set to 1 **if** (but not only if) the Y variables for sites p, q (missing entries in columns p, q) are set so that sites p and q become incompatible.

If M is an n by m matrix, then we have at most nm Y variables; $2m^2$ B variables; $m^2/2$ C variables; and $O(nm^2)$ inequalities in worst-case.

Finally, we have the objective function:

$$\text{Minimize } \sum_{(p,q) \text{ in } P} C(p, q)$$

Where P is the set of site-pairs that could be made to be incompatible.

Empirically, these ILPs solve very quickly (CPLEX 9) in fractions of seconds, or seconds

even for $m = n = 100$ and percentage of missing values up to 30%.

Data was generated with recombination and homoplasy by the program MS and modifications of MS.

Details are in COCOON 2007, Gusfield, Frid, Brown

Moreover, the ILP solution imputes the missing data with 2% - 5% error rate and improves as nm increases.

Extensions from Problem M1

- > Problem R1. Site-Removal Problem for **complete** data: Remove the **minimum** number of sites from the data, so that **no** incompatibilities remain. This is a common approach to incompatible data. NP-hard problem.
- > Problem S1. Site-Removal Problem with **missing** data: Impute values for the missing entries to minimize the solution to the resulting Site-Removal Problem.

ILP for S1 - a simple extension to the ILP for M1

- For each site i , let $D(i)$ be a variable set to 1 if and only if site i is removed.
- For each site-pair p,q in P , add the inequality $D(p) + D(q) - C(p,q) \Rightarrow 0$ to the M1 formulation. If sites p,q are incompatible (depending on how the missing values are set), then one of them must be removed.

Change the objective function to

Minimize $\sum_i D(i)$

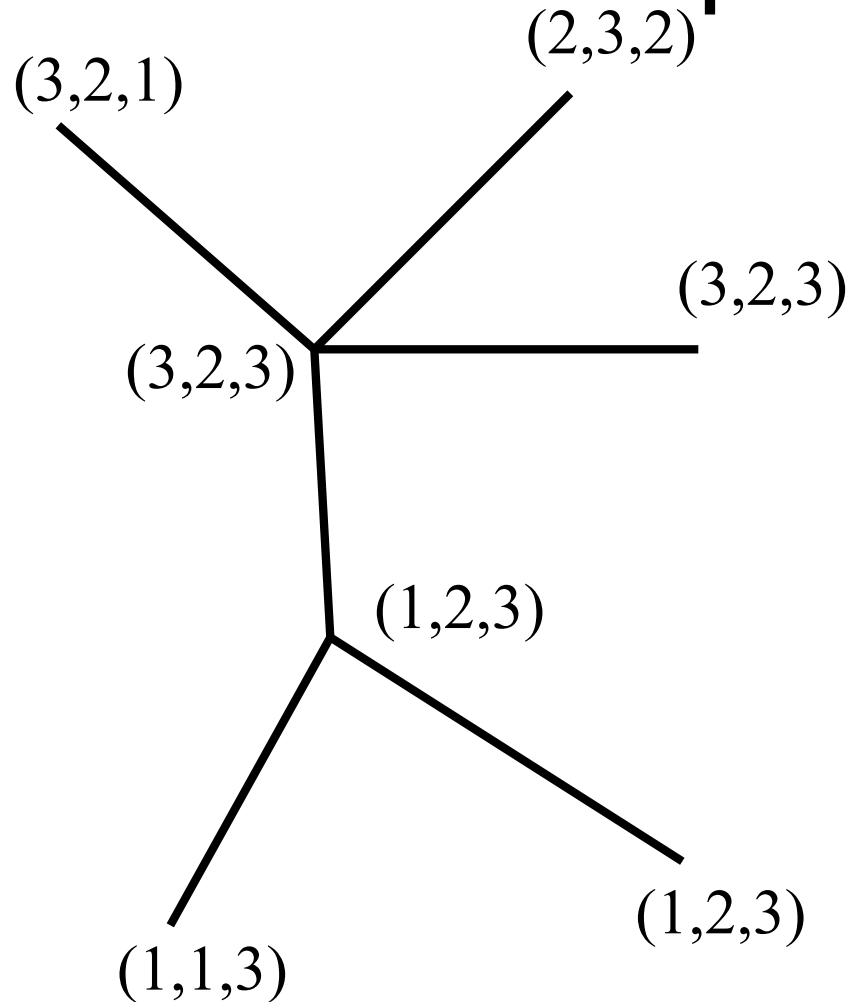
Extension to non-binary characters

We detail the case of three and four allowed states per character. The approach to the case of four states generalizes to any fixed number of allowed states per character, but the efficiency is expected to degrade rapidly with increasing number of states.

What is a Perfect Phylogeny for non-binary characters?

- Input consists of n sequences M with m sites (characters) each, where each site can take one of k values (states).
- In a Perfect Phylogeny T for M , each node of T is labeled with an m -length, k -ary sequence.
- T has n leaves, one for each sequence in M , labeled by that sequence.
- For **each** character-state pair (C,s) , the nodes of T that are labeled with state s for character C , form a **connected** subtree of T . It follows that the subtrees for any C are node-disjoint

Example: A perfect phylogeny for input M



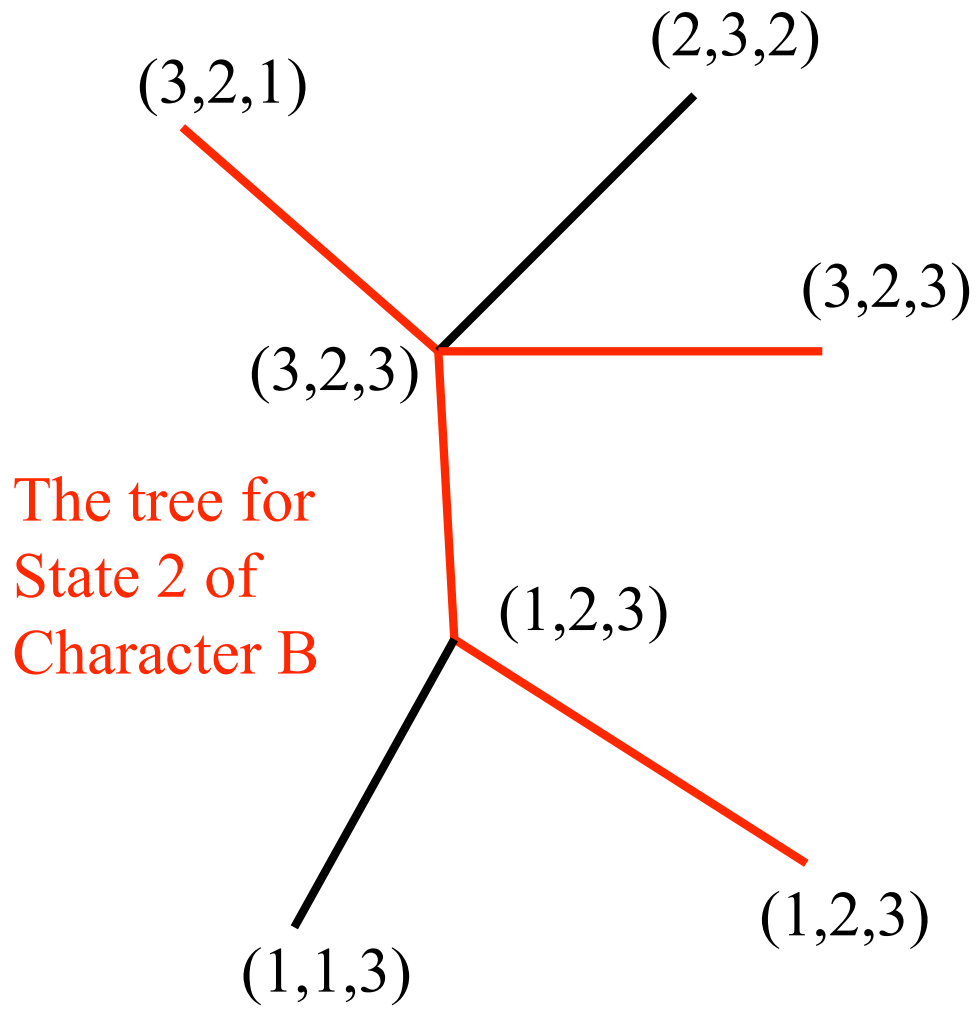
	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M

$$n = 5$$

$$m = 3$$

$$k = 3$$



	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M

$$n = 5$$

$$m = 3$$

$$k = 3$$

Three problems

- TR1: Given a ternary (1,2,3) matrix M , Remove the fewest sites (characters) of M so that the resulting matrix has a 3-state perfect phylogeny.
- TS1: Given a ternary matrix M with some ?s, Set each ? to 1,2, or 3 to minimize the solution to Problem TR1.
- Existence Problem: Is there a way to set the ?s so there is a 3-state perfect phylogeny?

Dress-Steel solution for 3-state Perfect phylogeny given **complete** data (1991)

- Recode each site $M(i)$ of M as three binary sites $M'(i,1)$, $M'(i,2)$, $M'(i,3)$ each indicating the taxa that have state 1, 2, or 3.
- Theorem (DS) There is a 3-state perfect phylogeny for M , if and only if there is a 2-state perfect phylogeny for some subset of M' consisting of exactly two of the columns $M'(i,1)$, $M'(i,2)$, $M'(i,3)$, for each column i of M .

Example

M

	A	B	C
1	3	2	1
2	2	3	2
3	3	2	3
4	1	1	3
5	1	2	3

M'

	A,1	A,2	A,3	B,1	B,2	B,3	C,1	C,2	C,3
1	0	0	1	0	1	0	1	0	0
2	0	1	0	0	0	1	0	1	0
3	0	0	1	0	1	0	0	0	1
4	1	0	0	1	0	0	0	0	1
5	1	0	0	0	1	0	0	0	1

↑ ↑ ↑ ↑ ↑ ↑

Compatible subset

ILP for the DS solution

$S(i,1)$, $S(i,2)$, $S(i,3)$ are binary variables indicating which columns of M' associated with column i in M will be selected. Then we need inequalities

$$S(i,1) + S(i,2) + S(i,3) = 2$$

$$S(i,x) + C(i,x; j,y) + S(j,y) \leq 2 \text{ etc. for } x,y = \{1,2,3\}, \text{ and}$$

$C(i,x;j,y)$ is the variable (essentially from the M1 ILP) that is forced to 1 if columns (i,x) and (j,y) in M' are incompatible.

From the DS theorem, the ILP is feasible if and only if there is a 3-state perfect phylogeny for M .

Handling missing values: When there is a ? in cell (p,q) of M , we use binary variables $Y(p,q,1)$, $Y(p,q,2)$, $Y(p,q,3)$ to indicate their values in M' , and add the equality:

$$Y(p,q,1) + Y(p,q,2) + Y(p,q,3) = 1$$

which sets the ? in cell (p,q) of M to either 1,2, or 3.

The resulting ILP is feasible if and only if the ?s in M have been set to allow a 3-state perfect phylogeny. That ILP solves the Existence Problem for three states per character.

To solve problem TR1: **minimize** the number of columns of M to remove, so that there is a 3-state solution, use variable $R(i)$ to indicate whether column i of M will be Retained. Then modify the DS ILP:

$S(i,1) + S(i,2) + S(i,3) \leq 2$ instead of $=$
and add $S(i,1) + S(i,2) + S(i,3) - 2R(i) \Rightarrow 0$

so that $R(i)$ can be set to 1 only if two of the three columns $M'(i,1)$, $M'(i,2)$, $M'(i,3)$ have been selected. Finally, use the objective function:

Maximize $\sum_{i \in M} R(i)$

Problem TS1

To solve Problem TS1, if there is a ? in cell (p,q) of M, we add the equality

$Y(p,q,1) + Y(p,q,2) + Y(p,q,3) = 1$ to the formulation for TR1.

Empirical Results: The 3-state Existence Problem

	0%	5%	10%	20%	35%	missing values
50 by 25, 3PP exists	0.0098	0.3	0.6	1.16	56.0	seconds
100 by 50 3PP exists	0.03	4.0	6.9	13.9	2,492.0	seconds

Times for data where no 3-state Perfect Phylogeny exists were similar, but smaller!

Empirical Results: Problems TR1 and TS1 (avg of 100 sets)

	TR1		TS1		
	0%	1%	5%	15%	missing values
30 by 100, r = 1	0.065	3.3	13.0	49.7	seconds
	93.7	93.8	94.12	94.4	# sites remaining

100 by 50

r = 3

0.17 seconds

40.18 # sites remaining

100 by 50

r = 5

0.2558 seconds

36.68 # of sites remaining

Parameter r influences the “closeness” of the data to a tree.

4-state perfect phylogeny

Problem FR1: Given quaternary sequences M , find the fewest number of sites to remove so that the resulting data M' has a 4-state perfect phylogeny.

Existence Problem: Is there a 4-state perfect phylogeny for M ?

We follow the Kannan-Warnow (1991) high-level idea for the existence problem as an ILP: Choose one of a finite set (between two and five, depending on the choice of ILP implementations) of possible tree types for each site and the generic splits that define each tree type; then require that the set of chosen splits be pairwise compatible. The ILP for Problem FR1 is built on the ILP for the 4-state Existence Problem.

Initial Empirical Results for 4-states (cplex 10)

Existence Problem:

30 by 30 0.2 secs for data with a 4-state PP
r = 1 0.49 secs for data with no 4-state PP

50 by 25
r = 3 0.24 secs for data with a 4-state PP
 0.27 secs for data with no 4-state PP

Initial* Empirical Results for 4-states (cplex 10)

Problem FR1 (avg. over 100 datasets)

50 by 25 r = 0	0.23 secs. 25 remaining sites	this might solve faster as an existence problem
50 by 25 r = 1	0.7 secs. 23.02 remaining sites	*These times are smaller than in the workshop talk, due to subsequent improvements in the ILP formulation.
50 by 25 r = 3	1.46 secs. 20.68 remaining sites	
100 by 50 r = 5	29.8 secs. 44 remaining sites	

Beyond 4 states

The approach used in the 4-state solution generalizes to any fixed number of states, but the times will increase as the number of states does. We are currently implementing five states.

Software

Perl script to generate the ILPs (for input to cplex or other ILP solvers) can be found at:

www.csif.cs.ucdavis.edu/~gusfield

The COCOON 2007 paper on the 2-state case is there also.