**Topic: Subtree Prune and Regraft**

Maximum Agreement Forest (MAF)

Forest contains the smallest number of components.

**Lemma 1:** $\text{dspr}(T, T') = |\text{MAF}(T, T')| - 1$

**Proof:**

1) $\text{dspr}(T, T') \geq |\text{MAF}(T, T')| - 1$

Assume: $T \rightarrow T'$ using dspr spr operations

Claim: $|\text{MAF}| \leq \text{dspr} + 1$

2) $\text{dspr} \leq |\text{MAF}| - 1$

Using Construction:

![Diagram showing the process of building a tree incrementally using MAF]

build the tree incrementally using MAF

Suppose we have our tree $T$:

![Diagram showing the process of adding T2 to make and grow the tree]
Case where this doesn't work:

| MAF | = 2, but we can't transform with one spr operation.  

To fix this add a special leaf $r$

Now $|MAF| = 3$

**Integer Linear Programming Approach**

Given $T$ and $T'$ find $MAF(T, T')$

Cut $T$ only and use $T'$ as a reference.

For every branch, we may choose to cut or not. Every edge has a boolean variable.

Def: $C_i = 1$ if the edge is cut
We want the min( \( \sum C_i \) ) such that:
1) Type 1: We ensure that if \( T_i \) from \( T \rightarrow T_i \) is a subtree of \( T' \)

Subtree means it can choose which edges to keep.

How can we tell two trees are topologically the same?

Use triples: \( i, j, k \) of 3 leaves!

- Two trees are topologically the same iff all triples are the same.

Consider all triples \((j,k,l)\) in \( T \) that are \textbf{NOT} the same in \( T \) and \( T' \)

Define \( S(j, k, l) \) as the set of edges that connect \( j, k, l \) in \( T \)

\[ \sum C_i \geq 1 \]

for \( i \) existing in \( S(j,k,l) \)

We need type 2 because they can overlap!
2) Type 2: Ensure no overlap

for all \((j,k)\) and \((p,q)\) non-overlap in \(T\) but overlap in \(T'\)

\[ \sum C_i \geq 1 \]

for \(I\) existing in path\((j,k)\) and path\((p,q)\)

**Fixed Parameter Tractable (FPT)**

**Parameters:** \(k\), size of \(n\) since we cannot solve this problem poly\((n)\)

Running time is \(O(2^k n)\)

\(k\) represents dspr

2004 Algorithm: \(O((56k)^{2^k}n^3)\)

2008 Algorithm: \(O(4^k k^4 + n^3)\)

2009 Algorithm: \(O(3^k n)\)

A more recent but more complicated algorithm runs in:

\(O(2.42^k n)\)

\(O(3^k n)\) Algorithm:

*sibling pairs*

MAF:

Shrink the tree!