Enumerating Minimum Path Covers of Trees

Merielyn Sher
Advisor: Dr. Charles Johnson

October 1, 2021

Outline

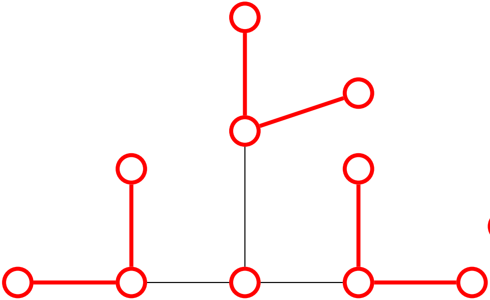
- Background
- Motivation
- Definitions
- Trees with Multiple-Component Hi-graphs
- Trees with Single-Component Hi-graphs
- Irreducible Trees

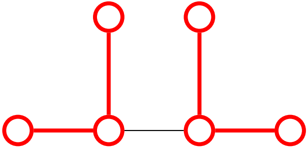
Background

- A **path cover** of a tree T is a collection of induced paths of T that are vertex disjoint and cover all the vertices of T.
- A **minimum path cover** (MPC) of T is a path cover with the minimum possible number of paths.
- The **path cover number** of T, denoted $P(T)$, is the number of paths in a MPC.
- $\mathcal{P}(T)$ denotes the set of all MPC's of T. $N(T)$ denotes the number of distinct MPC's of T. Note that $N(T) = |\mathcal{P}(T)|$.

Background

A tree can have unique or multiple MPC's.





$P(T_2) = 2; N(T_2) = 1$

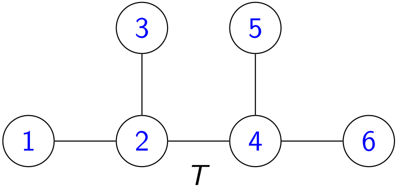
$P(T_1) = 4; N(T_1) = 19$

Motivation

- Let A $= (a_{ij})$ be an n $\times$ n Hermitian matrix. The graph of A, denoted G(A), is the simple undirected graph on n vertices with an edge $\{i, j\}$ if and only if $i \neq j$ and $a_{ij} \neq 0$.
- Given an undirected graph G, $\mathcal{H}(G)$ is the set of all Hermitian matrices with graph G.

$$\begin{array}{c c} & \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} & \begin{pmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 2 & 1 & 1 & 0 & 0 \\ 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 & 1 & 1 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{pmatrix} \end{array}$$

$$A \in \mathcal{H}(T)$$



Motivation

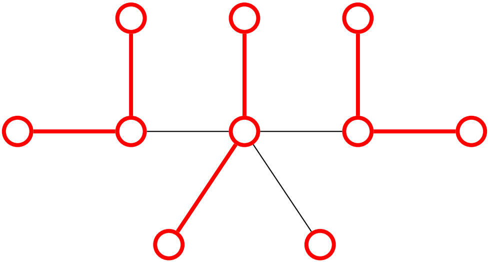
- For a matrix in $\mathcal{H}(G)$, the possible multiplicities of its eigenvalues are constrained by G.
- The **multiplicity list** of a $n \times n$ matrix is a simple partition of n in which the parts are the multiplicities of the distinct eigenvalues.
- For a graph G, a major constraint on the multiplicity lists for matrices in $\mathcal{H}(G)$ is the maximum multiplicity, $M(G)$, that is, the largest multiplicity that can occur.

Motivation

Theorem (JL-D99)

For any tree T, $M(T) = P(T)$.

Example:



Some possible multiplicity lists:

$$\{(4,3,1,1,1),(4,2,2,1,1),(4,2,1,1,1,1),(4,1,1,1,1,1,1),\cdots\}$$

$$M(T) = 4 = P(T).$$

Motivation

- Different ways that the maximum multiplicity can occur for a given tree
- $M(T) = P(T) \implies$ different ways that $P(T)$ occurs?

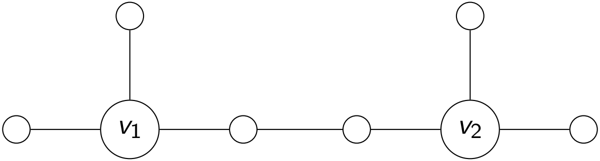
Definitions

- The **degree** of a vertex v in a tree T, denoted $\deg_T(v)$, is the number of neighboring branches at v
- A **high degree vertex** (HDV) in a tree is a vertex of degree at least 3. A **low degree vertex** has degree 1 or 2.

Definitions

The **Hi-graph**, $H(T)$, of a tree T is the subgraph induced by its HDV's. $H(T)$ is a forest with one or more components (each of which is a tree).

E.g.



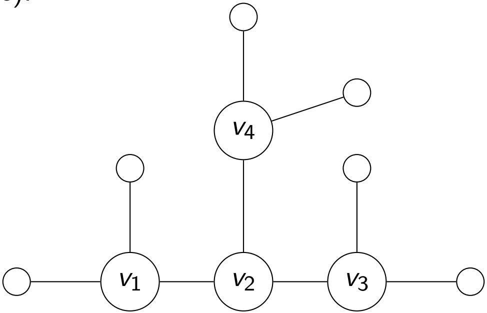
T

$H(T)$

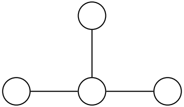
Definitions

The **Hi-graph**, $H(T)$, of a tree T is the subgraph induced by its HDV's. $H(T)$ is a forest with one or more components (each of which is a tree).

E.g.



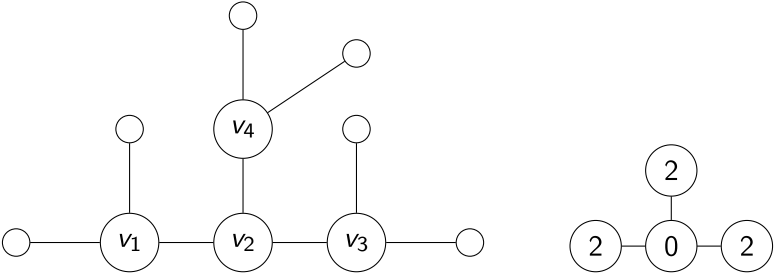
T



$H(T)$

Definitions

- The **incremental degree** of a vertex v in T, $\text{ideg}_T(v)$, is the difference between its degrees in T and in $H(T)$.
- A **high-incremental degree** (HID) vertex in $H(T)$ is one of incremental degree at least 2; otherwise it is of **low-incremental degree** (LID).
- A **labeled Hi-graph**, denoted $H_L(T)$, has all its vertices labeled with their respective incremental degrees.



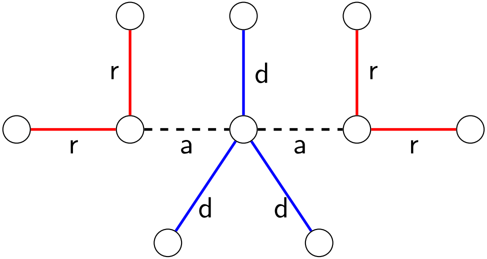
T $H_L(T)$

Edge Status

Definition

An edge is **absent** if it is used in no MPC of T; An edge is **required** if it is used in all MPC's; An edge is **discretionary** if it occurs in some but not all MPC's.

Example:

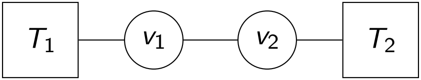


Edge Status

Proposition

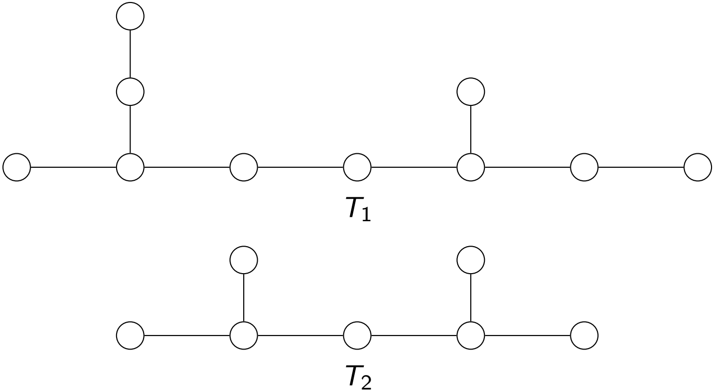
Any edge between two low degree vertices is a required edge.

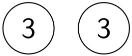
For a tree T, the value of $N(T)$ is independent of the lengths of the paths induced by the low degree vertices in T.



Theorem

If two trees T_1 and T_2 have the same labeled Hi-graph, then $P(T_1) = P(T_2)$ and $N(T_1) = N(T_2)$.





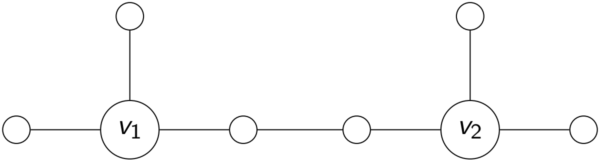
$H_L(T_1) \& H_L(T_2)$

$$P(T_1) = P(T_2) = 3; N(T_1) = N(T_2) = 9$$

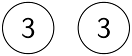
Trees with Multiple-Component Hi-graphs

Trees with Multiple-Component Hi-graphs

The Hi-graph of a tree T has two or more components when there are one or more low-degree vertices on a single path between two of the HDV's.



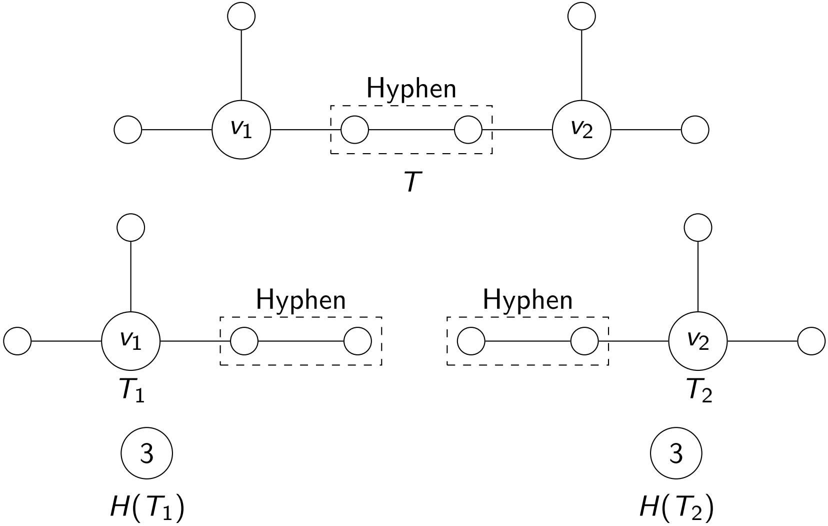
T



$H(T)$

Hyphen Decomposition

A **hyphen** is a path induced by the low-degree vertex or vertices between two HDV's in T.



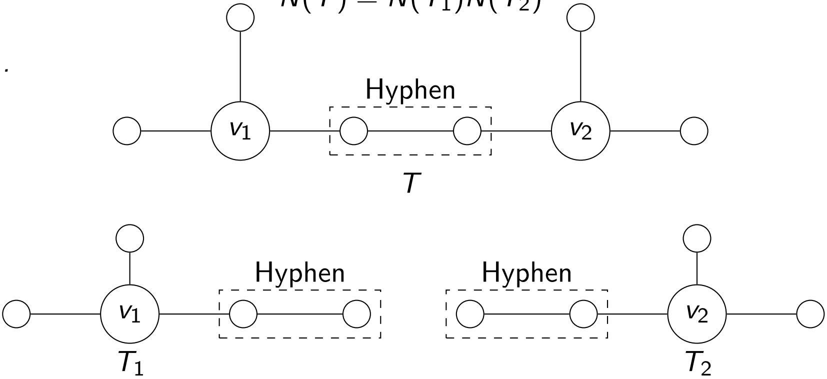
Trees with Multiple-Component Hi-graphs

Proposition

Let T be a tree with a two-component Hi-graph, and let T_1 and T_2 be the results of hyphen-decomposing T. Then,

$$N(T) = N(T_1)N(T_2)$$

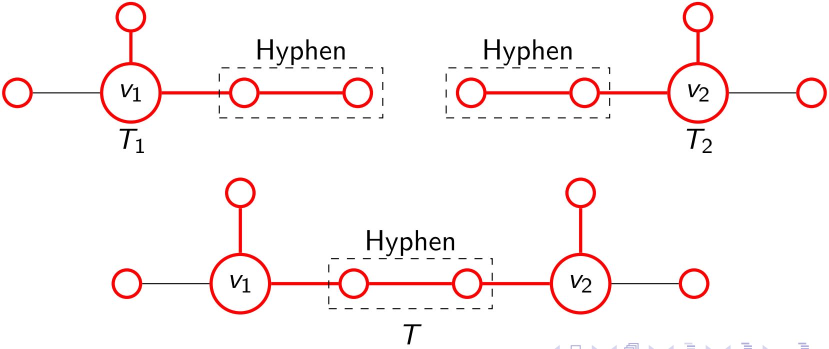
.



$$N(T_1) = 3; N(T_2) = 3; N(T) = 9$$

Proof idea:

- The hyphen in T is always included in a single path in every MPC of T_1 and T_2.
- For any two MPC's of T_1 and T_2, a MPC for T can be constructed by merging the two respective paths in the two MPC's that contain the hyphen.
- Construct a function $f : \mathcal{P}(T_1) \times \mathcal{P}(T_2) \rightarrow \mathcal{P}(T)$, and show that f is a bijection.
- Thus, $N(T_1) \times N(T_2) = |\mathcal{P}(T_1)||\mathcal{P}(T_2)| = |\mathcal{P}(T)| = N(T)$.



Trees with Multiple-Component Hi-graphs

Theorem

Suppose that the Hi-graph of a tree T consists of n disjoint components. If T is hyphen-decomposed into n disjoint components, $T_1, T_2, \cdots, T_n$, then,

$$N(T) = \prod_{i=1}^{n} N(T_i).$$

Trees with Single-Component Hi-graphs

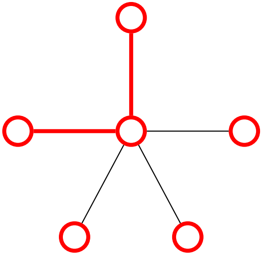
Trees with Single-Component Hi-graphs

A **generalized star**, or g-star, is a tree with at most one HDV.

Proposition

Let T be a g-star with k arms. Then,

$$P(T) = k - 1 \quad \textit{and} \quad N(T) = \binom{k}{2}.$$



$P(T) = 4; N(T) = \binom{5}{2} = 10$

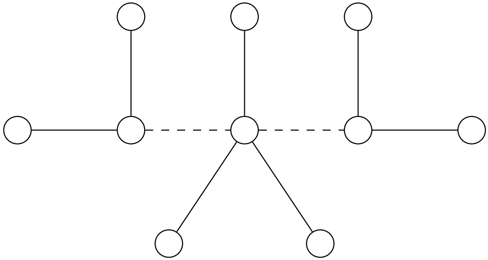
Absent Edge Decomposition

Theorem

If a tree T is decomposed into smaller components $T_1, T_2, \cdots, T_n$ through removing all of its absent edges, then,

$$P(T) = \sum_{i=1}^{n} P(T_i) \quad \textit{and} \quad N(T) = \prod_{i=1}^{n} N(T_i).$$

Example:

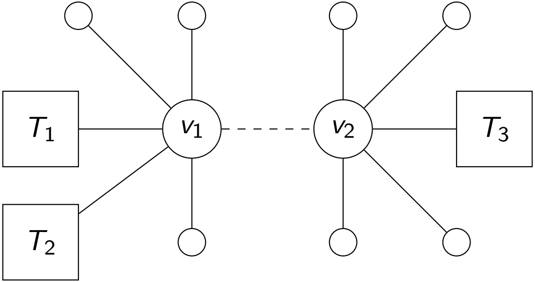


$$N(T) = 1 \times \binom{3}{2} \times 1 = 3$$

Identifying Absent Edges

Proposition

In a tree T, an edge between two HID vertices is an absent edge.



Not the only type of absent edges!

Trees with Single-Component Hi-graphs - Identifying Absent Edges

A HDV is **peripheral** if and only if starting from itself, there is at most one direction to proceed in T in order to find another HDV. A **pendent g-star** in a tree T is a g-star induced by a peripheral HDV and its pendent paths.

A tree T is either a g-star itself or contains two or more pendent g-stars.

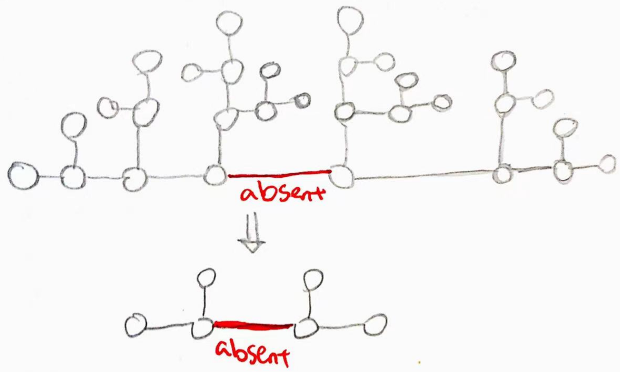
Proposition

Removing a pendent g-star (as well as the edge that connects it with the rest of T) from a tree T does not change the status of the rest of the edges in T.

Trees with Single-Component Hi-graphs - Identifying Absent Edges

"Pruning" Trees to Identify Absent Edges:

- For a tree T, select an edge e with a status that cannot be directly identified.
- Repeatedly remove pendent g-stars as well as the edges that connect them to the rest of the tree from T.
- Stop when
  (1) e is between two HID vertices (e is absent), or
  (2) a single g-star that contains e is left (e is either discretionary or required).


absent
absent

Irreducible Trees

Irreducible Trees

Definition

A tree T is **irreducible** if $H(T)$ is connected and there are no absent edges in T.

Lemma

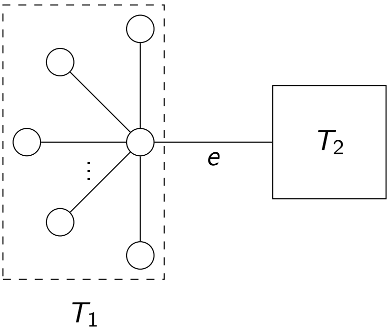
For any tree T, an edge e connecting a pendent g-star to the rest of T is never required.

Lemma

For an irreducible tree T, an edge e connecting a pendent g-star to the rest of T is discretionary.

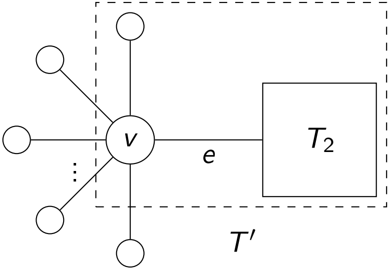
Irreducible Trees - An Algorithm to Enumerate MPC's

- Identify an edge e connecting a pendent g-star that has k pendent paths with the rest of T.
- Partition $\mathcal{P}(T)$ into two subsets where e is either always used or never used.
- For $\mathcal{P}_N(T)$, the set where e is never used, consider the two trees T_1 and T_2 as the result of the removal of e from T. We have $|\mathcal{P}_N(T)| = N(T_1)N(T_2) = \binom{k}{2}N(T_2)$.



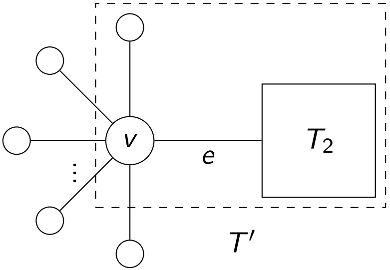
Irreducible Trees - An Algorithm to Enumerate MPC's

- Now consider $\mathcal{P}_U(T)$, where e is always used.
- In order to minimize the number of paths used, for every MPC in $\mathcal{P}_U(T)$, the path that includes e must also go through the central vertex of the pendent g-star as well as one of its pendent paths.
- We construct a subtree T' through selecting one of the k pendent paths at v and removing the other $(k-1)$ pendent paths from T. There are $\binom{k}{1} = k$ ways of doing this.



Irreducible Trees - An Algorithm to Enumerate MPC's

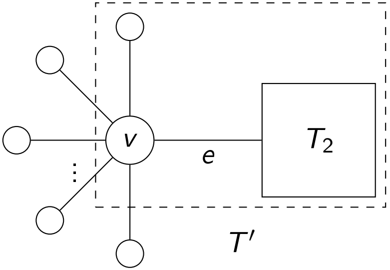
- For each of the k ways, we count the number of MPC's of T' that use e to be consistent with our setup.
- "Luckily", e is required in T', so $N(T')$ is exactly the number of MPC's of T' that use e.
- The resulting trees, regardless of which of the k pendent paths at v is selected, are all isomorphic to one another and thus have the same number of MPC's, $N(T')$.
- Therefore, $|\mathcal{P}_U| = k \times N(T')$.



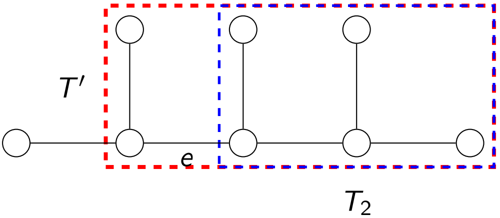
Irreducible Trees - An Algorithm to Enumerate MPC's

Finally, we have

$$N(T) = |\mathcal{P}(T)| = |\mathcal{P}_N(T)| + |\mathcal{P}_U(T)| = \binom{k}{2} \times N(T_2) + k \times N(T').$$



Example:



$$N(T) = 1 \times 3 + \binom{2}{1} \times 1 = 5$$

A Complete Process of Counting MPC's

1. Apply hyphen decomposition to decompose T into components with connected Hi-graphs.

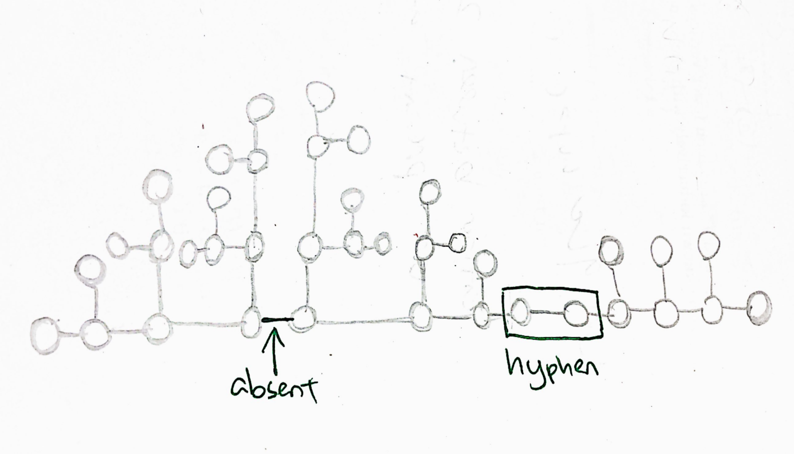
2. For each of the resulting components, identify all absent edges and apply absent edge decomposition.

3. For each of the resulting components, repeat steps 1 and 2 since new hyphens and absent edges can occur after the decomposition process. When the resulting components become irreducible, go to step 4.

4. Use the algorithm to calculate the number of MPC's for irreducible trees inductively.

5. Recombine the numbers to obtain $N(T)$.

An Example



$N(T) = 3125$

Thank you!

References

- C.R. Johnson and A. Leal Duarte, *The maximum multiplicity of an eigenvalue in a matrix whose graph is a tree*, Linear and Multilinear Algebra 46 (1999), 139-144.
- C.R. Johnson and A. Leal Duarte, *On the possible multiplicities of the eigenvalues of an Hermitian matrix whose graph is a given tree*, Linear Algebra Appl. 348 (2002), 7-21.
- C.R. Johnson and C.M. Saiago, *Eigenvalues, Multiplicities and Graphs*, Cambridge University Press, Cambridge, 2018.