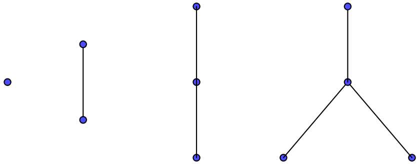
Star Covers and Star Partitions of Cographs and Butterfly-free Graphs

Joyashree Mondal S Vijayakumar

IIITDM KANCHEEPURAM

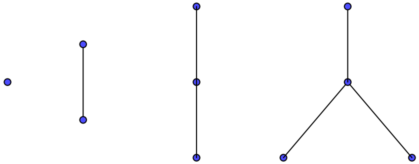
Star

A graph that is isomorphic to $K_{1,r}$, for some $r \geq 0$, is called a star.



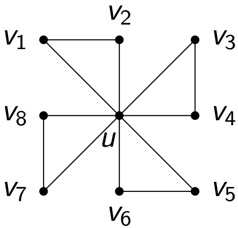
Star

A graph that is isomorphic to $K_{1,r}$, for some $r \geq 0$, is called a star.

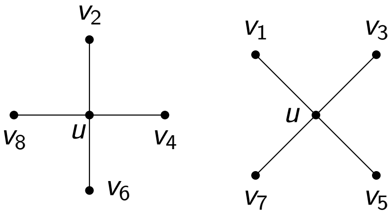


Note: Each star has **center**.

STAR in a Graph

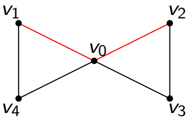


$G = F_4$



Star Cover of a Graph

Let $G = (V, E)$ be a graph. Then a collection $(V_1, \ldots, V_k)$ of subsets of V is called a star cover of G if each set in the collection induces a star and has $V_1 \cup \ldots \cup V_k = V$.



G

$(\{v_0, v_1, v_2\}, \{v_0, v_3, v_4\})$

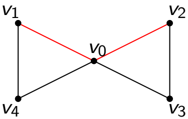
A Star Cover of G

The Star Cover Number, $sc(G)$

The minimum k for which a graph G admits a star cover $(V_1, \ldots, V_k)$ is called the star cover number of G and is denoted by $sc(G)$.

Star Partition of a Graph

A star cover $(V_1, \ldots, V_k)$ of a graph $G = (V, E)$ is called a star partition if $(V_1, \ldots, V_k)$ is a partition of V.



G

$$(\{v_0, v_1, v_2\}, \{v_3\}, \{v_4\})$$

A Star Partition of G

The Star Partition Number, $sp(G)$

The minimum k for which a graph G admits a star partition $(V_1, \ldots, V_k)$ is called the star partition number of G and is denoted by $sp(G)$.

Problems

STAR COVER

Input: A graph G.
Goal: A star cover of G of minimum size.

STAR PARTITION

Input: A graph G.
Goal: A star partition of G of minimum size.

Star Cover vs Star Partition

The problems are similar but not the same!

Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$

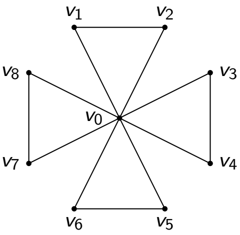
Star Cover vs Star Partition

The problems are similar but not the same!

Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4

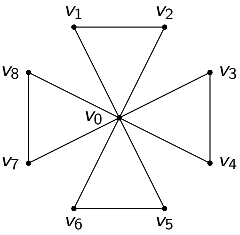
Star Cover vs Star Partition

The problems are similar but not the same!

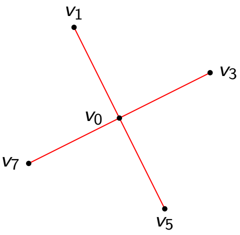
Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4



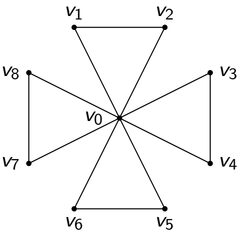
Star Cover vs Star Partition

The problems are similar but not the same!

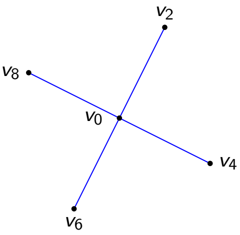
Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4



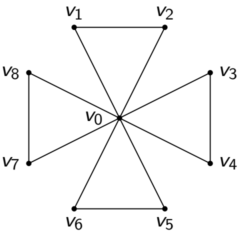
Star Cover vs Star Partition

The problems are similar but not the same!

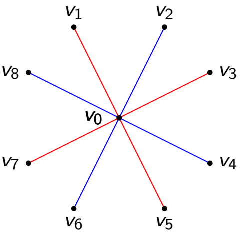
Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4



A Star Cover of F_4

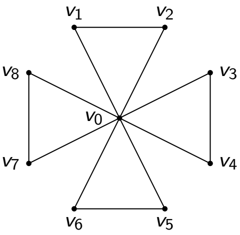
Star Cover vs Star Partition

The problems are similar but not the same!

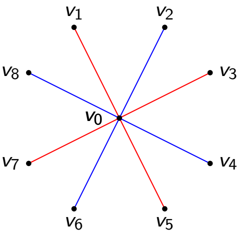
Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

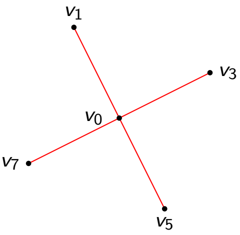
$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4



A Star Cover of F_4



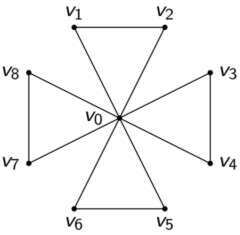
Star Cover vs Star Partition

The problems are similar but not the same!

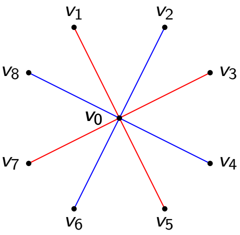
Example

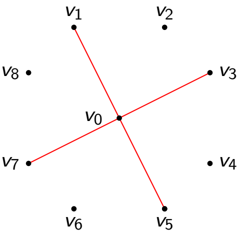
For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4





A Star Cover of F_4

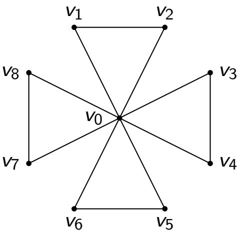
Star Cover vs Star Partition

The problems are similar but not the same!

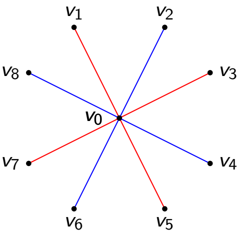
Example

For $n \geq 2$, let $F_n = K_1 \oplus nK_2$ be the friendship graph. Then

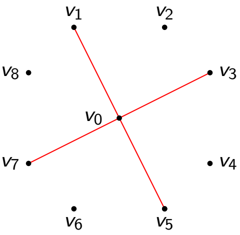
$$sc(F_n) = 2 \quad \text{but} \quad sp(F_n) = n + 1.$$



F_4



A Star Cover of F_4



A Star Partition of F_4

Literature Survey

NP-Completeness Results

Star Cover and Star Partition are NP-hard for

- Chordal bipartite graphs [4]
- C_4-free bipartite graphs [4]
- Bipartite planar graphs [6, 1]
- $K_{1,5}$-free split graphs [1]
- Line graphs [2, 1]
- Co-tripartite graphs [3, 1].

Also deciding whether an input graph can be covered by *or* partitioned into three stars is NP-complete [1].

Literature Survey

Polynomial Time Algorithms

STAR COVER and STAR PARTITION have polynomial time algorithms for

- bipartite permutation graphs [3, 5]
- convex bipartite graphs [1, 1]
- doubly-convex bipartite graphs [1]
- trees [4]
- claw-free split graphs [our result]
- double-split graphs [our result].

Literature Survey

Approximation and Inapproximation Results

- Star Partition has a polynomial time $r/2$-approximation algorithm for $K_{1,r}$-free graphs [1, 2].
- Star Cover and Star Partition have a polynomial time
  - A 2-approximation algorithm for split graphs [1];
  - $O(\log n)$-approximation algorithms for triangle-free graphs [2];
  - $(d+1)$-approximation algorithm for triangle-free graphs of degree at most d [2].
- It is NP-hard to approximate Star Partition within $n^{1/2-\epsilon}$ for all $\epsilon > 0$ [1, 3].
- Star Cover and Star Partition do not have any polynomial time $c \log n$-approximation algorithm for some constant $c > 0$ unless $\mathsf{P} = \mathsf{NP}$ [2].

HEREDITARY GRAPHS

Hereditary Graphs

Definition

A graph class is called a hereditary graph class if for every graph in the class it also includes all its induced subgraphs.

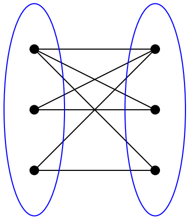
Hereditary Graphs

Definition

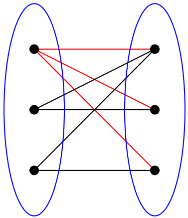
A graph class is called a hereditary graph class if for every graph in the class it also includes all its induced subgraphs.

Examples: $\mathcal{F}$-free graphs. For instance, triangle-free graphs, chordal graphs and, more generally, perfect graphs are hereditary graphs.

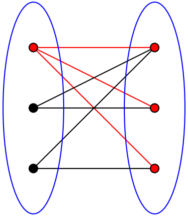
Hereditary graphs: An Algorithm for Star Cover



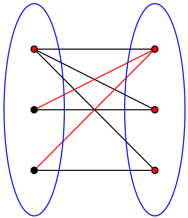
Hereditary graphs: An Algorithm for Star Cover



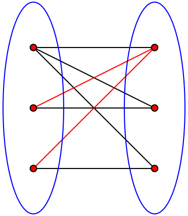
Hereditary graphs: An Algorithm for Star Cover



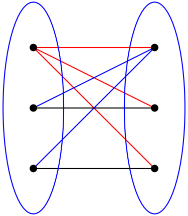
Hereditary graphs: An Algorithm for Star Cover



Hereditary graphs: An Algorithm for Star Cover



Hereditary graphs: An Algorithm for Star Cover



Hereditary graphs: An Algorithm for Star Cover

Algorithm: *Approximate-hereditary*

Input: A graph G from the hereditary graph class $\mathcal{G}$.
Output: A star cover $\mathcal{S}$ of G.

1. Set $\mathcal{S} = \emptyset$.
2. Colour all vertices of G black.
3. While G has a black vertex repeat the following:
   - Find a star Z with a maximum number of black vertices.
   - Colour the black vertices of G that are in Z red.
   - Set $\mathcal{S} = \mathcal{S} \cup \{Z\}$.
4. Output $\mathcal{S}$.

Hereditary graphs: An Approximation Algorithm for Star Cover

Our solution size: $apx\text{-}sc(G) \leq (c \log n) \cdot sc(G)$.

Hereditary graphs: An Approximation Algorithm for Star Cover

Our solution size: $\textit{apx-sc}(G) \leq (c \log n) \cdot sc(G)$. (Follows from the *greedy* set cover algorithm.)

Hereditary graphs: An Approximation Algorithm for Star Cover

Our solution size: $\textit{apx-sc}(G) \leq (c \log n) \cdot \textit{sc}(G)$. (Follows from the *greedy* set cover algorithm.)

So, this is an $O(\log n)$-approximation algorithm.

Hereditary graphs: An Approximation Algorithm for Star Cover

Our solution size: $apx\text{-}sc(G) \leq (c \log n) \cdot sc(G)$. (Follows from the *greedy* set cover algorithm.)

So, this is an $O(\log n)$-approximation algorithm.

Theorem

Star Cover has an $O(n^2 t(n))$ time $O(\log n)$-approximation algorithm for any hereditary graph class for which the maximum independent set can be computed in $O(t(n))$ time.

Hereditary graphs: An Approximation Algorithm for Star Cover

Our solution size: $apx\text{-}sc(G) \leq (c \log n) \cdot sc(G)$. (Follows from the *greedy* set cover algorithm.)

So, this is an $O(\log n)$-approximation algorithm.

Theorem

STAR COVER has an $O(n^2 t(n))$ time $O(\log n)$-approximation algorithm for any hereditary graph class for which the maximum independent set can be computed in $O(t(n))$ time.

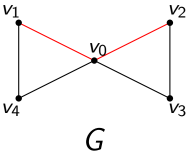
Corollary

STAR COVER has a polynomial time $O(\log n)$-approximation algorithm for perfect graphs and the $O(\log n)$ approximation factor can not be improved assuming $P \neq NP$ [a].

[a] V.V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin Heidelberg (2001)

BUTTERFLY-FREE GRAPHS

The Butterfly Graph



$sc(G) = 2$

$sp(G) = 3$

Butterfly-free graphs includes-

- Bipartite graphs (in fact, all triangle-free graphs).
- Split graphs (in fact, all $2K_2$-free graphs).

Note: If G is a triangle-free graph then $sp(G) = sc(G) = \gamma(G)$.

Theorem

For any butterfly-free graph G, $sp(G) = sc(G)$. Moreover, given a star cover of G, a star partition of at most the same size can be computed in time $O(n^2 \log n)$.

Theorem

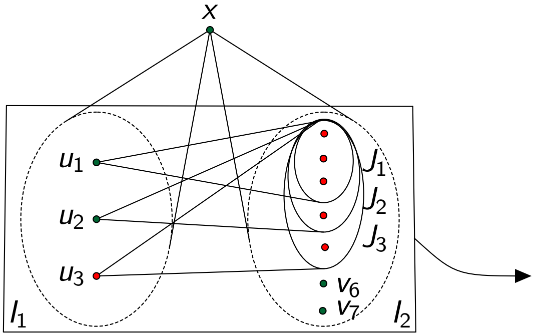
For any butterfly-free graph G, $sp(G) = sc(G)$. Moreover, given a star cover of G, a star partition of at most the same size can be computed in time $O(n^2 \log n)$.

Proof idea: Let (x, l_1) and (x, l_2) be two stars in an optimal star cover of G.

Theorem

For any butterfly-free graph G, $sp(G) = sc(G)$. Moreover, given a star cover of G, a star partition of at most the same size can be computed in time $O(n^2 \log n)$.

Proof idea: Let (x, I_1) and (x, I_2) be two stars in an optimal star cover of G.

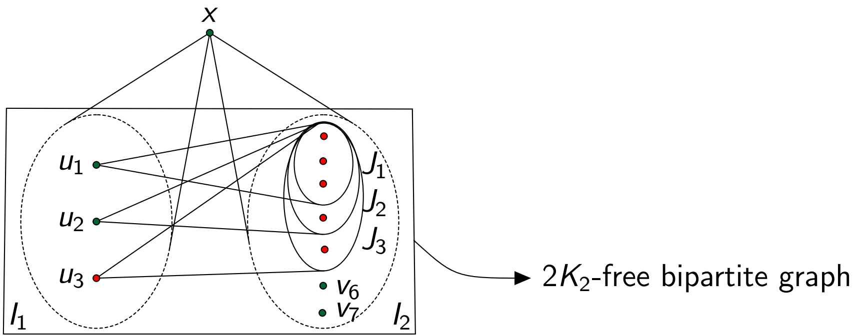


2K_2-free bipartite graph

Theorem

For any butterfly-free graph G, $sp(G) = sc(G)$. Moreover, given a star cover of G, a star partition of at most the same size can be computed in time $O(n^2 \log n)$.

Proof idea: Let (x, I_1) and (x, I_2) be two stars in an optimal star cover of G.



Two stars (x, I_1) and (x, I_2) can be replaced with other two stars $(x, \{u_1, u_2, v_6, v_7\})$ and (u_3, J_3).

Butterfly-free graphs: Approximation algorithm

- Butterfly-free graphs form a hereditary graph class.

[1]Farber, M. (1989), On diameters and radii of bridged graphs, Discrete Math., 73(3), 249–260.

Butterfly-free graphs: Approximation algorithm

- Butterfly-free graphs form a hereditary graph class.
- The maximum independent set problem has an $O(n^4)$ time exact algorithm for this graph class [1].

[1] Farber, M. (1989), On diameters and radii of bridged graphs, Discrete Math., 73(3), 249–260.

Butterfly-free graphs: Approximation algorithm

- Butterfly-free graphs form a hereditary graph class.
- The maximum independent set problem has an $O(n^4)$ time exact algorithm for this graph class [1].
- We have proved: $sp(G) = sc(G)$ for any butterfly-free graph G.

[1] Farber, M. (1989), On diameters and radii of bridged graphs, Discrete Math., 73(3), 249–260.

Butterfly-free graphs: Approximation algorithm

- Butterfly-free graphs form a hereditary graph class.
- The maximum independent set problem has an $O(n^4)$ time exact algorithm for this graph class [1].
- We have proved: $sp(G) = sc(G)$ for any butterfly-free graph G.
- The approximation algorithm for Star Cover on hereditary graph classes now imply the following theorem.

[1] Farber, M. (1989), On diameters and radii of bridged graphs, Discrete Math., 73(3), 249–260.

Theorem

Both STAR COVER *and* STAR PARTITION *have an $O(n^6)$ time $O(\log n)$-approximation algorithm for* STAR PARTITION *on butterfly-free graphs. Moreover the $O(\log n)$ approximation factor can not be improved.*

Theorem

Both STAR COVER *and* STAR PARTITION *have an* $O(n^6)$ *time* $O(\log n)$*-approximation algorithm for* STAR PARTITION *on butterfly-free graphs. Moreover the* $O(\log n)$ *approximation factor can not be improved.* [a]

[a] V.V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin Heidelberg (2001)

COGRAPHS

Cographs

Cographs

A graph is called a cograph if it is P_4-free.

Fact

A graph is a cograph if and only if it can be obtained from K_1's by a finite number of union and join operations.

Note

The bottom-up construction of a cograph is often represented by a binary tree, namely its *cotree*.

Results on Cographs

Both Star Cover and Star Partition have:

- $O(n^2)$ time algorithms for trivially perfect graphs ((C_4, P_4)-free graphs).
- $O(n^2)$ time algorithms for co-trivially perfect graphs ($(2K_2, P_4)$-free graphs).
- Linear time algorithms for threshold graphs ($(C_4, 2K_2, P_4)$-free graphs).

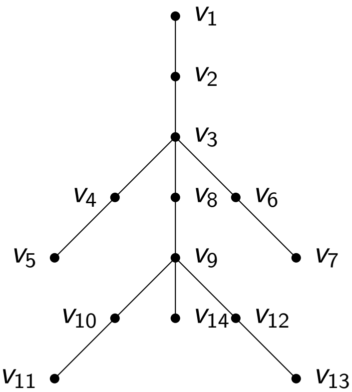
TRIVIALLY PERFECT GRAPHS

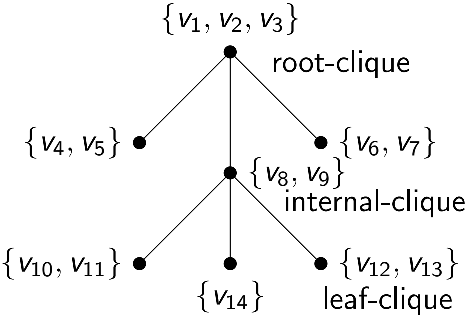
((C_4, P_4)-free Graphs)

Trivially Perfect Graphs: Clique Tree Representation

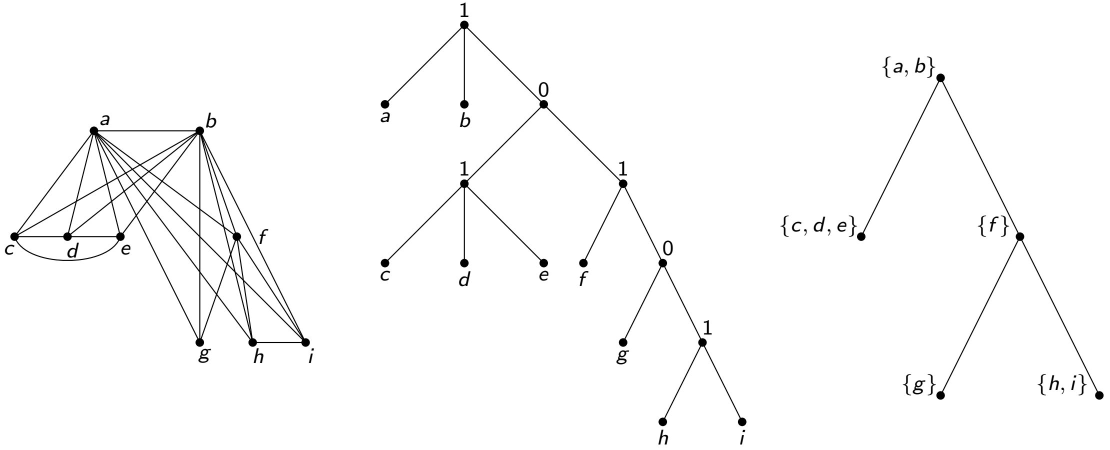
Connected trivially perfect graphs are comparability graphs of rooted trees.

Clique trees *are compressed forms of the underlying rooted trees.*





Construction of a Clique Tree



1

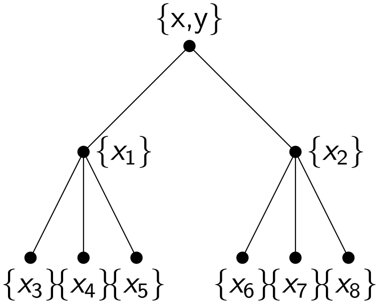
[1] D.G. Corneil, Y. Perl and L.K. Stewart, A linear recognition algorithm for cographs, SIAM Journal on Computing, 14 (4) (1985) 926-934.

Lemma

Let $G \ncong K_n$ be a connected trivially perfect graph and suppose that it is given by its clique tree representation. Then any *optimal star cover (partition) of G has a star with the center* alone *from the root-clique of G.*

Proof

- let $\{x, y\}$ be any star from R and let $\{x_1\} \cup I_1$ and $\{x_2\} \cup I_2$ be two stars from two different components of $V \setminus R$.
- These three starts can be replaced by the two stars $(\{x\} \cup \{x_1, x_2\}$ and $\{y\} \cup (I_1 \cup I_2)$ of G.



$$I_1 = \{x_3, x_4, x_5\}$$

$$I_2 = \{x_6, x_7, x_8\}$$

Note:

Let $X = \{x\} \cup I$ be any maximum star of a connected trivially perfect graph G that is not a complete graph. Then the following hold:

1. The set I is a maximum independent set of G and consists of exactly one vertex from each of the leaf-cliques of the clique-tree $T(G)$ so that the center x *alone* belongs to the root-clique of $T(G)$.
2. Any star Z of G of size more than two has its center necessarily from some internal clique of $T(G)$.

Lemma

Let G be any connected trivially perfect graph that is not a complete graph. Then G has an optimal star cover (partition) containing some maximum star of G.

A Theorem Leading to Greedy Algorithms

Theorem

Let G be any connected trivially perfect graph. Then any maximum star of G belongs to some optimal star cover (partition) of G.

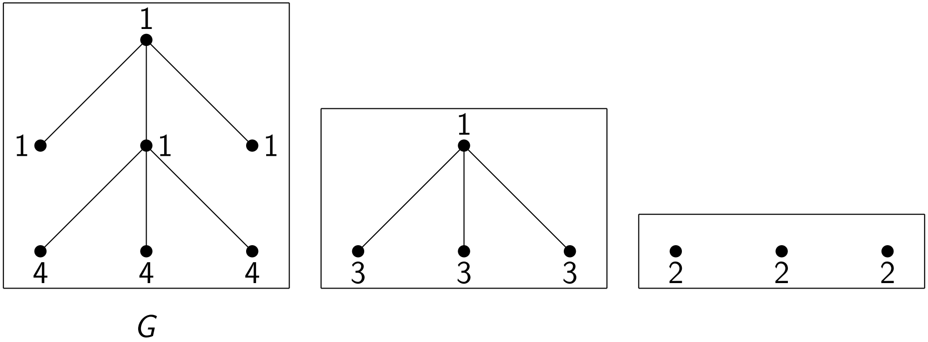
Algorithm for Star Partition on Trivially Perfect Graphs

Input: A trivially perfect graph G.
Output: A star partition $\mathcal{C}$ of G.

1. Set $\mathcal{C} = \emptyset$.
2. While G is not the null graph, repeat the following:
   - Pick a component H of G.
   - Find a maximum star X of H.
   - Set $\mathcal{C} = \mathcal{C} \cup \{X\}$.
   - Set $G = G \setminus X$.
3. Output $\mathcal{C}$.

Trivially Perfect Graphs

Execution of the Algorithm for Star Partition



Therefore $sp(G) = 5$.

Theorem

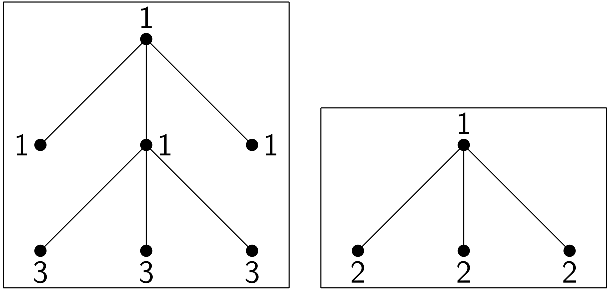
STAR PARTITION *has an* $O(n^2)$ *time exact algorithm for trivially perfect graphs.*

STAR COVER on Trivially Perfect Graphs

Theorem

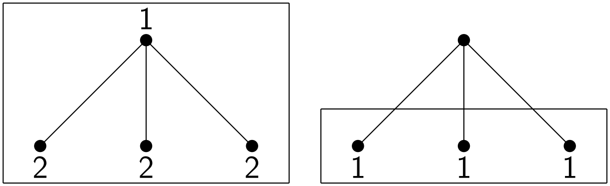
Let $G = (V, E)$ be any connected trivially perfect graph and let $X = \{v\} \cup I$ be any maximum induced star of G. Let $H = G \setminus I$ if $G \setminus X$ is disconnected and let $H = G \setminus X$ otherwise. Let $\mathcal{C}'$ be any optimal star cover of H. Then $\mathcal{C} = \{X\} \cup \mathcal{C}'$ is an optimal star cover of G.

When $G \setminus X$ is Connected



G

When $G \setminus X$ is Not Connected



G

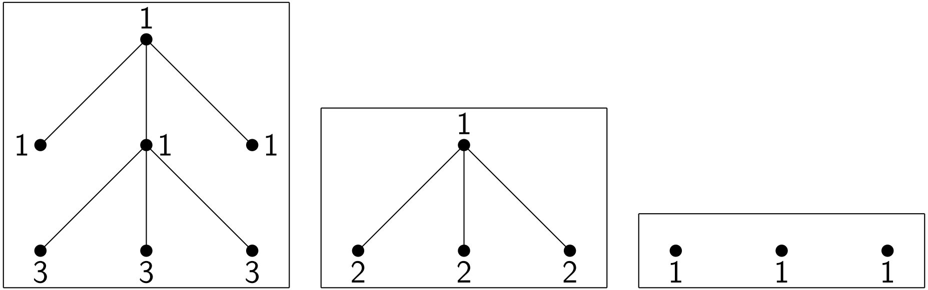
Algorithm for STAR COVER on Trivially Perfect Graphs

Input: A trivially perfect graph G.
Output: A star cover $\mathcal{C}$ of G.

1. Set $\mathcal{C} = \emptyset$.
2. While G is not the null graph, repeat the following:
   - Pick a component H of G.
   - Find a maximum star $X = \{x\} \cup I$ of H.
   - Set $\mathcal{C} = \mathcal{C} \cup \{X\}$.
   - If $H \setminus X$ is disconnected, set $G = G \setminus I$. Else set $G = G \setminus X$.
3. Output $\mathcal{C}$.

Trivially Perfect Graphs

Execution of the Algorithm for Star Cover

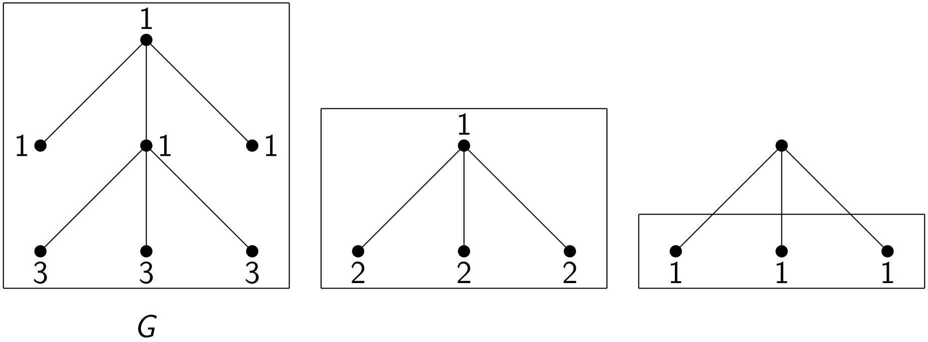


G

Therefore $sc(G) = 3$.

Trivially Perfect Graphs

Execution of the Algorithm for Star Cover



Therefore $sc(G) = 3$.

Theorem

STAR COVER *has an* $O(n^2)$ *time exact algorithm for trivially perfect graphs.*

CO-TRIVIALLY PERFECT GRAPHS

($(2K_2, P_4)$-free Graphs)

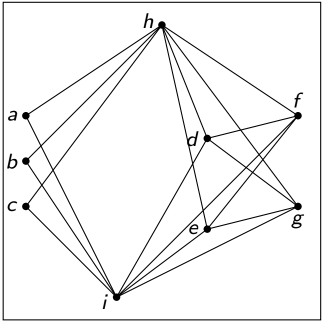
Equivalence of Star Cover and Star Partition on Co-Trivially Perfect Graphs

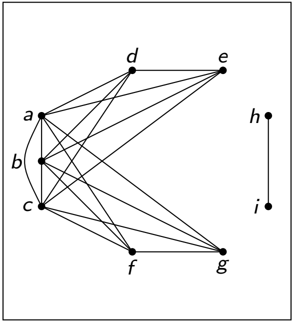
Fact

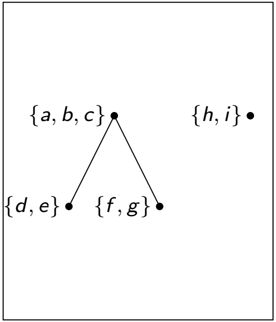
If G is a co-trivially perfect graph, then $sp(G) = sc(G)$.

Implication: Suffices to study any one of Star Cover and Star Partition.

Co-clique Forest Representation of $(2K_2, P_4)$-free Graphs

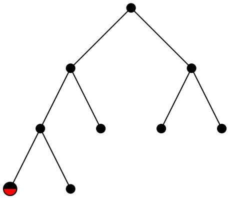




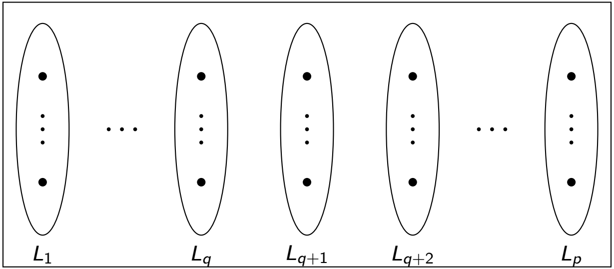


Lemma

Let G be a connected $(2K_2, P_4)$-free graph. Suppose that G is given by its co-clique forest representation. Then G has a minimum star partition $\mathcal{C}$ such that the centers of stars in $\mathcal{C}$ are from the bottom most nodes of G.



Complete Multipartite Graphs



G

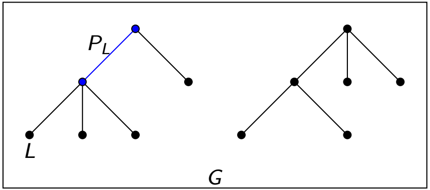
Proposition

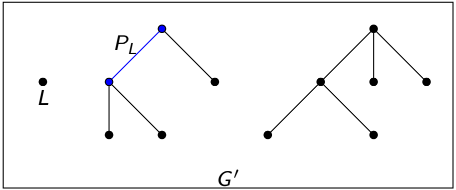
Let $G = (L_1, \ldots, L_p)$ be a complete multipartite graph with $|L_1| \leq \ldots \leq |L_p|$ and let q be the largest integer such that $|L_1 \cup L_2 \cup \ldots \cup L_q| \leq p - q$. Then $sp(G) = p - q$.

$(2K_2, P_4)$-free Graphs: A Lower Bound on $sp(G)$

Lemma

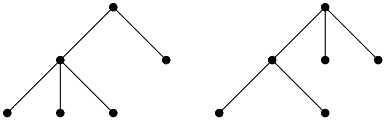
Let $G \not\cong K_1$ be a connected $(2K_2, P_4)$-free graph. Suppose that G is given by its co-clique forest representation. Let L be a leaf node of smallest size in G and let $G' = L \oplus (G \setminus L)$. Then $sp(G') \leq sp(G)$.



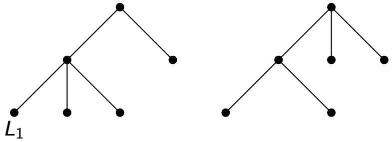


Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph

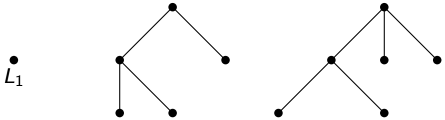
Let G be a $(2K_2, P_4)$-free graph.



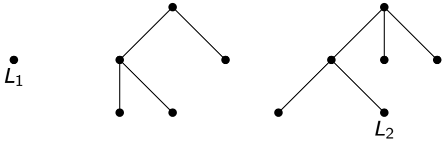
Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



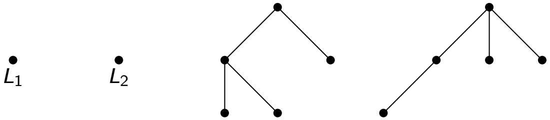
Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



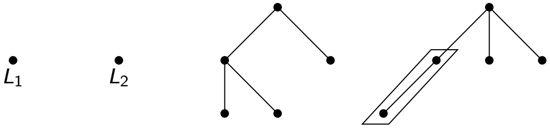
Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



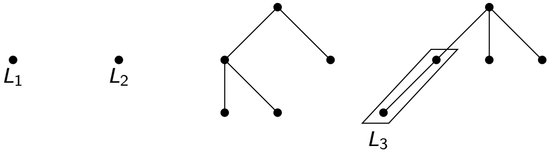
Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



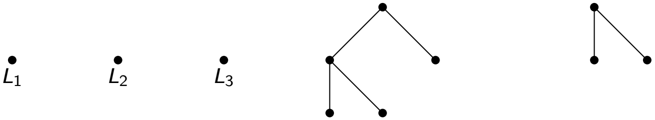
Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph



Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph

And let G be transformed into the complete multipartite graph G'.

G'

Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph

And let G be transformed into the complete multipartite graph G'.

G'

- By repeated application of the preceding lemma, $sp(G') \leq sp(G)$.

Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph

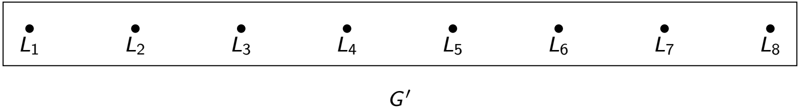
And let G be transformed into the complete multipartite graph G'.

G'

- By repeated application of the preceding lemma, $sp(G') \leq sp(G)$.
- Let p be the number of leaves in G.
- Let q be the maximum value such that $|L_1| + \cdots + |L_q| \leq p - q$.
- Then $sp(G) \geq sp(G') = p - q$.

Transformation of a $(2K_2, P_4)$-free Graph to a Specific Complete Multipartite Graph

And let G be transformed into the complete multipartite graph G'.



- By repeated application of the preceding lemma, $sp(G') \leq sp(G)$.
- Let p be the number of leaves in G.
- Let q be the maximum value such that $|L_1| + \cdots + |L_q| \leq p - q$.
- Then $sp(G) \geq sp(G') = p - q$.

Hence the lower bound for Star Partition on $(2K_2, P_4)$-free graphs is $p - q$.

The Algorithm

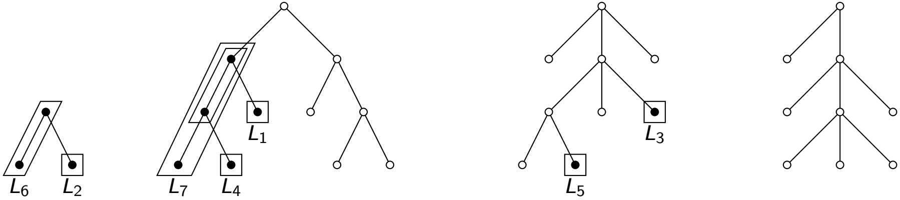
Algorithm: *Co-Trivially Perfect*

Input: A connected $(2K_2, P_4)$-free graph in its co-clique forest representation.
Output: A star partition of G.

1. Compute the complete multipartite graph $G' = (L_1, \ldots, L_p)$ from G by repeatedly removing the least size leaf node and making it a stand-alone co-clique tree.
2. Find the largest integer q such that $|L_1 \cup L_2 \cup \ldots \cup L_q| \leq p - q$.
3. Color the vertices of G that are in $L_1 \cup \ldots \cup L_q$ **black** and others **white**.
4. *Compute* and output a star partition of G of size $p - q$ $(p - q + 1)$ using vertices in $L_1 \cup \ldots \cup L_q$ as centers.

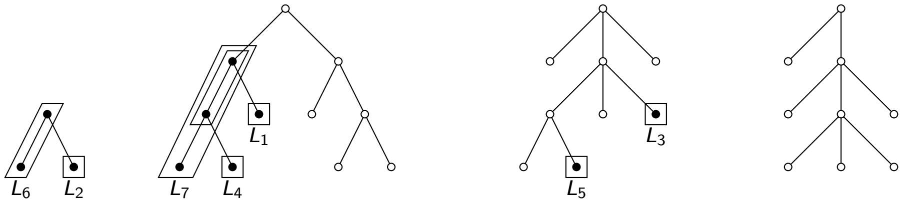
Properties of Black and White Nodes

- If an internal node is black, then all of its children are black.
- If an internal node is white but has a black child, then it also has a white child.



Properties of Black and White Nodes

- If an internal node is black, then all of its children are black.
- If an internal node is white but has a black child, then it also has a white child.

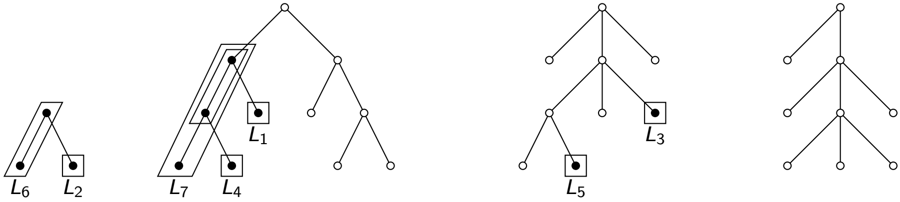


- Partition the co-clique forest G into sets W, B and M:
  - W consists of trees with only white nodes.
  - B consists of trees with only black nodes.
  - M consists of trees with both white and black nodes.

Computing the Star Partition

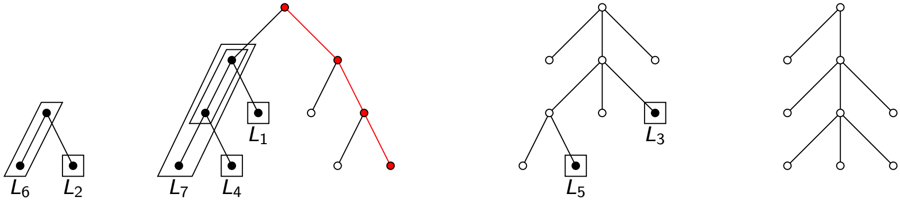
1. While $M \neq \phi$, form stars and separate black nodes.
2. While $B \neq \phi$, form stars with black vertices as centers and white vertices as independent part.
3. While $W \neq \phi$, partition remaining white co-clique forest into stars.

Algorithm with an example



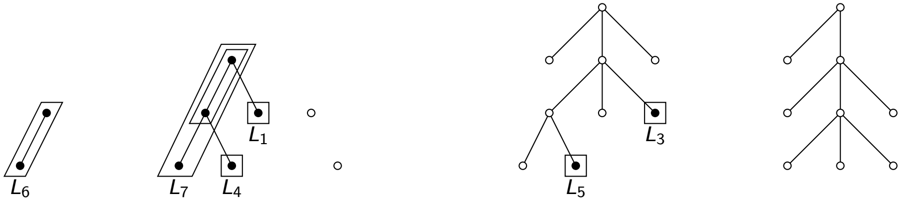
- $M \neq \phi$.
- $B \neq \phi$.
- $W \neq \phi$.

Algorithm with an example



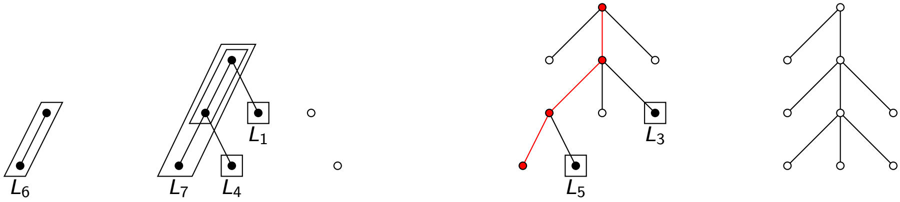
- $M \neq \phi$.
- $B \neq \phi$.
- $W \neq \phi$.

Algorithm with an example



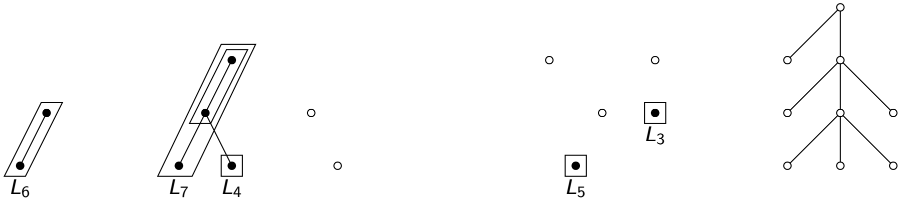
- $M \neq \phi$.
- $B \neq \phi$.
- $W \neq \phi$.

Algorithm with an example



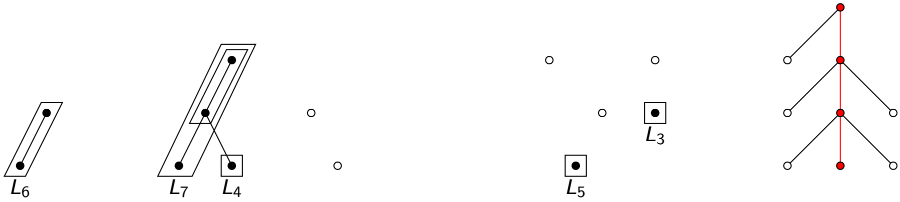
- $M \neq \phi$.
- $B \neq \phi$.
- $W \neq \phi$.

Algorithm with an example



- $B \neq \phi$.
- $W \neq \phi$.
- $M = \phi$.

Algorithm with an example



- $B \neq \phi$.
- $W \neq \phi$.
- $M = \phi$.

Algorithm with an example

- $B \neq \phi$.
- $W \neq \phi$.
- $M = \phi$.

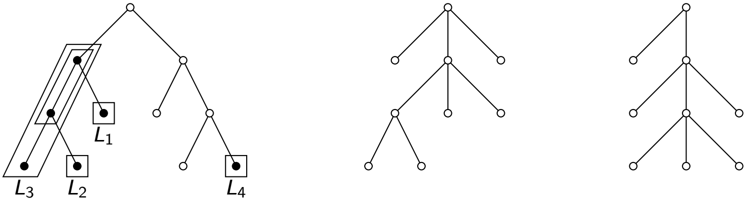
Algorithm with an example

- $W \neq \phi$.
- $M = \phi$.
- $B = \phi$.
- Now, number of stars $= |L_1| + \cdots + |L_q| = s$.

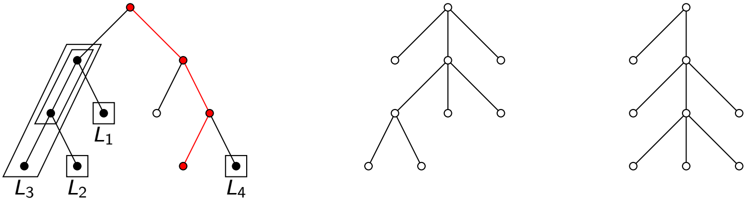
Algorithm with an example

- $W \neq \phi$.
- $M = \phi$.
- $B = \phi$.
- Now, number of stars $= |L_1| + \cdots + |L_q| = s$.
- Hence the solution size $= s + ((p - q) - s) = p - q$.

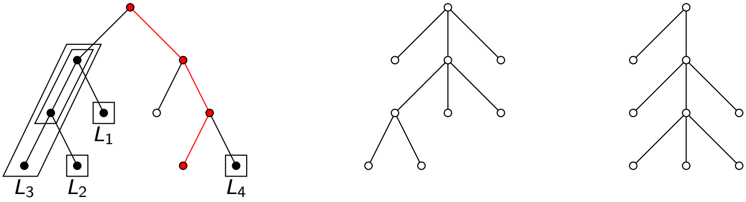
All Black Nodes From Same Co-clique Tree



All Black Nodes From Same Co-clique Tree

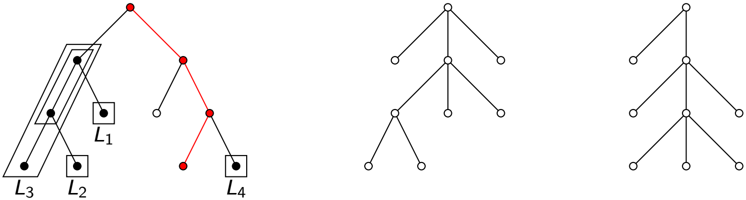


All Black Nodes From Same Co-clique Tree



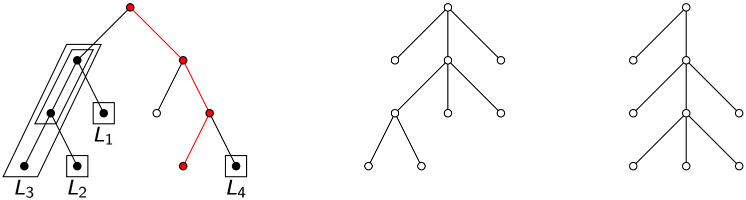
- For this path we will use a white vertex from other tree.

All Black Nodes From Same Co-clique Tree



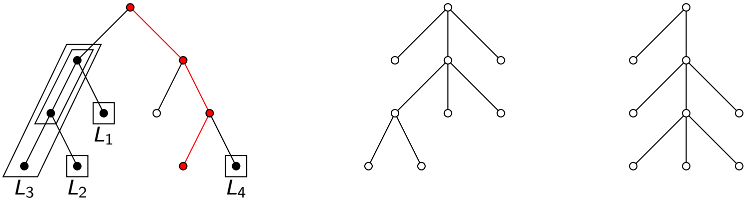
- For this path we will use a white vertex from other tree.
- Let $s = |L_1| + \cdots + |L_q|$.

All Black Nodes From Same Co-clique Tree



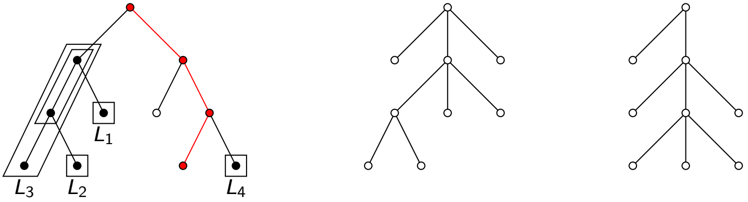
- For this path we will use a white vertex from other tree.
- Let $s = |L_1| + \cdots + |L_q|$.
- When $B = \phi$, number of stars $= s + 1$.

All Black Nodes From Same Co-clique Tree



- For this path we will use a white vertex from other tree.
- Let $s = |L_1| + \cdots + |L_q|$.
- When $B = \phi$, number of stars $= s + 1$.
- If $s = (p - q)$, then number of stars $= p - q + 1$.

All Black Nodes From Same Co-clique Tree



- For this path we will use a white vertex from other tree.
- Let $s = |L_1| + \cdots + |L_q|$.
- When $B = \phi$, number of stars $= s + 1$.
- If $s = (p - q)$, then number of stars $= p - q + 1$.
- If $s < (p - q)$, then number of stars $= (s + 1) + ((p - q) - (s + 1)) = p - q$.

Lemma

Let G be a connected $(2K_2, P_4)$-free graph. If $L_1, \ldots, L_q$ are all necessarily from the same tree T such that $(T \setminus (L_1 \cup \cdots \cup L_q)) \neq \phi$ and $|L_1| + \cdots + |L_q| = p - q$, then $sp(G) > p - q$.

Proof.

- Suppose $sp(G) = p - q$.
- This implies at least q bottom parts of G must be used fully as centers.
- This is possible only if the $(p - q) = s$ centers are coming from T.
- Also $|T| > p - q$.
- This implies at least the vertices from root node of T are non-centers.
- Hence $sp(G) > p - q$.

□

Theorem

- *Let G be a connected $(2K_2, P_4)$-free graph and suppose that it is given by its co-clique tree representation.*
- *Let $G' = (L_1, \ldots, L_p)$ be the corresponding complete multipartite graph.*
- *Let q be the largest integer such that $|L_1| + \cdots + |L_q| \leq p - q$.*

If $L_1, \ldots, L_q$ are all necessarily from the same tree T of size more than $|L_1 \cup \cdots \cup L_q|$ and $|L_1| + \cdots + |L_q| = p - q$, then $sp(G) = p - q + 1$. Else $sp(G) = p - q$.

THRESHOLD GRAPHS

$((C_4, 2K_2, P_4)$-free Graphs)

A Linear Time Algorithm for Threshold Graphs

Theorem

Let G be a connected threshold graph. Then $sp(G) = \lceil \omega(G)/2 \rceil$. Indeed G can be partitioned into a clique and at most one star.

Moreover, an optimal star partition of G can be computed in linear time.

Future Scope

1. Determine the computational complexity of Star Cover and Star Partition for cographs.

References

- J. Bang-Jensen, J. Huang, G. MacGillivray, A. Yeo, Domination in convex bipartite and convex-round graphs. Technical Report, University of Southern Denmark, 1999.
- A. Björklund, T. Husfeldt, M. Koivisto, Set partitioning via inclusion-exclusion, SIAM Journal on Computing 39 (2009) 543-563.
- A. Brandstädt and D. Kratsch, On the restriction of some NP-complete graph problems to permutation graphs, in L. Budach, ed., *Proc. of the FCT '85 Conf.*, Lecture Notes in Computer Science 199 (1985) 53-62.
- E.J. Cockayne, S. Goodman, S.T. Hedetniemi, A linear algorithm for the domination number of a tree, Inform. Process. Lett. 4 (1975), 41-44.
- Corneil, D. G. and Perl, Y. and Stewart, L. K., A Linear Recognition Algorithm for Cographs, SIAM Journal on Computing, Volume 14 (1985), 926-934.

- P. Damaschke, H. Müller, and D. Kratch, Domination in convex and chordal bipartite graphs, Information Processing Letters 36 (1990) 231 - 236.
- D. Dor and M. Tarsi, Graph Decomposition is NP-Complete: A Complete Proof of Holyer's Conjecture, SIAM J. Comput., 26(4) (1997), 1166-1187.
- R.G. Downey and M.R. Fellows, Fundamentals of Parameterized Complexity. Undegraduate texts in computer science. Springer (2013)
- O. Duginov, Partitioning the vertex set of a bipartite graph into complete bipartite subgraphs, Discrete Mathematics and Theoretical Computer Science 16:3 (2014) 203-214.
- M. Farber and J.M. Keil, Domination in permutation graphs, J. Algorithms 6 (1985) 309-321.
- M. R. Garey , D. S. Johnson, Computers and Intractability; A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., New York, 1990.

- M.C. Golumbic, Algorithmic Graph Theory and Perfect Graphs, 2nd Edition, Vol. 57, Elsevier, 2004.
- A.K. Kelmans, Optimal packing of stars in a graph, Discrete Mathematics 173, (1997) 97-127.
- F. Maffray, M. Preissmann, On the NP-completeness of the k-colorability problem for triangle-free graphs, Discrete Mathematics 162 (1996), 313-317.
- H. Müller and A. Brandstädt, The NP-completeness of Steiner tree and dominating set for chordal bipartite graphs, Theoretical Computer Science 53 (1987) 257-265.
- V. Raman, S. Saurabh, Short cycles make W-hard problems hard: FPT algorithms for W-hard problems in graphs with no short cycles. Algorithmica 52 (2008), 203-225.

- M.A. Shalu, S. Vijayakumar, T.P. Sandhya, J. Mondal, Induced star partition of graphs. Discrete Applied Mathematics 319 (2022), 81-91.
- V.V. Vazirani, Approximation Algorithms, Springer-Verlag, Berlin Heidelberg (2001)
- D. Zuckerman, Linear degree extractors and the inapproximability of Max Clique and Chromatic Number, Theory of Computing, Volume 3 (2007), 103-128.