Enumeration of Minimal Connected Vertex Covers and Dominating Sets

Dieter Kratsch
Université de Lorraine, France

Joint work with Petr Golovach and Pinar Heggernes

GROW 2015, Aussois, October 11-15, 2015

Abstract

Abstract

On general graphs

We show that a graph on n vertices has at most $O(1.8668^n)$ minimal connected vertex covers and these can be listed in time $O(1.8668^n)$.

Abstract

On general graphs

We show that a graph on n vertices has at most $O(1.8668^n)$ minimal connected vertex covers and these can be listed in time $O(1.8668^n)$.

However

We are not able to find a graph with more than $3^{(n-1)/3} \approx 1.4422^n$ minimal connected vertex covers.

Abstract

On general graphs

We show that a graph on n vertices has at most $O(1.8668^n)$ minimal connected vertex covers and these can be listed in time $O(1.8668^n)$.

However

We are not able to find a graph with more than $3^{(n-1)/3} \approx 1.4422^n$ minimal connected vertex covers.

On graph classes

We show that graphs of chordality ≤ 5 have at most $O(1.6181^n)$ minimal connected vertex covers.

Abstract

On general graphs

We show that a graph on n vertices has at most $O(1.8668^n)$ minimal connected vertex covers and these can be listed in time $O(1.8668^n)$.

However

We are not able to find a graph with more than $3^{(n-1)/3} \approx 1.4422^n$ minimal connected vertex covers.

On graph classes

We show that chordal graphs and distance hereditary graphs have at most $O(3^{n/3})$ minimal connected vertex covers.

Enumeration algorithms

Enumeration in graphs

Given a property $\mathcal{P}$, we want to list all the distinct vertex subsets satisfying $\mathcal{P}$ in a given graph G on n vertices.

Enumeration algorithms

Enumeration in graphs

Given a property $\mathcal{P}$, we want to list all the distinct vertex subsets satisfying $\mathcal{P}$ in a given graph G on n vertices.

Trivial algorithm

If property $\mathcal{P}$ can be tested in polynomial time, we can check for every subset of vertices of G whether it satisfies $\mathcal{P}$ and output it if so in time $O(2^n \cdot poly(n))$.

Enumeration algorithms

Enumeration in graphs

Given a property $\mathcal{P}$, we want to list all the distinct vertex subsets satisfying $\mathcal{P}$ in a given graph G on n vertices.

Trivial algorithm

If property $\mathcal{P}$ can be tested in polynomial time, we can check for every subset of vertices of G whether it satisfies $\mathcal{P}$ and output it if so in time $O(2^n \cdot poly(n))$.

More clever branching algorithms

- Recursive algorithms with branching and reduction rules.
- Running time very often gives an upper bound on the number of objects that a graph can have.

Minimal (connected) vertex covers

A set of vertices X in a graph G is a vertex cover if every edge of G has an endpoint in X.

Minimal (connected) vertex covers

A set of vertices X in a graph G is a vertex cover if every edge of G has an endpoint in X.

X is a connected vertex cover if X is a vertex cover and $G[X]$ is connected.

Minimal (connected) vertex covers

A set of vertices X in a graph G is a vertex cover if every edge of G has an endpoint in X.

X is a connected vertex cover if X is a vertex cover and $G[X]$ is connected.

A (connected) vertex cover X is minimal if no proper subset of X is a (connected) vertex cover.

Minimal (connected) vertex covers

A set of vertices X in a graph G is a vertex cover if every edge of G has an endpoint in X.

X is a connected vertex cover if X is a vertex cover and $G[X]$ is connected.

A (connected) vertex cover X is minimal if no proper subset of X is a (connected) vertex cover.

A connected vertex cover X is minimal if every $x \in X$ is either a cut vertex of $G[X]$ or it has a private edge.

Vertex covers in general

Observation

- *X is a vertex cover of $G = (V, E)$ if and only if $V \setminus X$ is an independent set of G.*

Vertex covers in general

Observation

- *X is a vertex cover of $G = (V, E)$ if and only if $V \setminus X$ is an independent set of G.*
- *If a vertex v is not in a vertex cover X then all neighbors of v are in X.*

A classical example

Theorem (Moon and Moser; 1965)

Every graph on n vertices has at most $3^{n/3} < 1.4423^n$ minimal vertex covers,

A classical example

Theorem (Moon and Moser; 1965)

Every graph on n vertices has at most $3^{n/3} < 1.4423^n$ minimal vertex covers, and there are graphs that have $3^{n/3} < 1.4423^n$ minimal vertex covers,

A classical example

Theorem (Moon and Moser; 1965)

Every graph on n vertices has at most $3^{n/3} < 1.4423^n$ minimal vertex covers, and there are graphs that have $3^{n/3} < 1.4423^n$ minimal vertex covers, and the minimal vertex covers of a graph can be enumerated in $O(1.4423^n)$ time.

A classical example

Theorem (Moon and Moser; 1965)

Every graph on n vertices has at most $3^{n/3} < 1.4423^n$ minimal vertex covers, and there are graphs that have $3^{n/3} < 1.4423^n$ minimal vertex covers, and the minimal vertex covers of a graph can be enumerated in $O(1.4423^n)$ time.

The union of $n/3$ triangles has $3^{n/3}$ maximal independent sets and thus also $3^{n/3}$ minimal vertex covers.

Motivation

- The problem of computing a minimum connected vertex cover is as old as computing a minimum vertex cover.

Motivation

- The problem of computing a minimum connected vertex cover is as old as computing a minimum vertex cover.
- The number of vertex covers of a graph and the enumeration of these form pioneering examples of enumeration algorithms.

Motivation

- The problem of computing a minimum connected vertex cover is as old as computing a minimum vertex cover.
- The number of vertex covers of a graph and the enumeration of these form pioneering examples of enumeration algorithms.
- The maximum number of minimal **connected** vertex covers of a graph or the enumeration of these has not been studied.

Motivation

- The problem of computing a minimum connected vertex cover is as old as computing a minimum vertex cover.
- The number of vertex covers of a graph and the enumeration of these form pioneering examples of enumeration algorithms.
- The maximum number of minimal **connected** vertex covers of a graph or the enumeration of these has not been studied.
- Enumeration and maximum number of minimal vertex covers have been studied on graph classes. On triangle-free graphs there are better (tight) bounds.

Motivation

- The problem of computing a minimum connected vertex cover is as old as computing a minimum vertex cover.
- The number of vertex covers of a graph and the enumeration of these form pioneering examples of enumeration algorithms.
- The maximum number of minimal **connected** vertex covers of a graph or the enumeration of these has not been studied.
- Enumeration and maximum number of minimal vertex covers have been studied on graph classes. On triangle-free graphs there are better (tight) bounds.
- Computation of minimum sets is not a part of our motivation.

Minimal connected vertex covers of general graphs

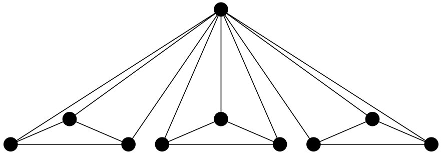
Theorem

The maximum number of minimal connected vertex covers of an arbitrary graph is $O(1.8668^n)$*, and these can be enumerated in time* $O(1.8668^n)$*.*

Minimal connected vertex covers of general graphs

Theorem

The maximum number of minimal connected vertex covers of an arbitrary graph is $O(1.8668^n)$, and these can be enumerated in time $O(1.8668^n)$.



A graph with $3^{(n-1)/3} \approx 1.4422$ minimal connected vertex covers.

Sketch of the proof

A recursive branching algorithm:

Sketch of the proof

A recursive branching algorithm:

Let G be an input graph. Given two disjoint vertex subsets of G:

- set S of selected vertices
- set F of free vertices

Sketch of the proof

A recursive branching algorithm:

Let G be an input graph. Given two disjoint vertex subsets of G:

- set S of selected vertices
- set F of free vertices

Algorithm $\textsc{EnumCVC}(S, F)$ generates all minimal connected vertex covers X of G such that $S \subseteq X \subseteq S \cup F$.

Sketch of the proof

A recursive branching algorithm:

Let G be an input graph. Given two disjoint vertex subsets of G:

- set S of selected vertices
- set F of free vertices

Algorithm $\textsc{EnumCVC}(S, F)$ generates all minimal connected vertex covers X of G such that $S \subseteq X \subseteq S \cup F$.

Initial call: $\textsc{EnumCVC}(\emptyset, V(G))$.

Sketch of the proof

A recursive branching algorithm:

Let G be an input graph. Given two disjoint vertex subsets of G:

- set S of selected vertices
- set F of free vertices

Algorithm $\textsc{EnumCVC}(S, F)$ generates all minimal connected vertex covers X of G such that $S \subseteq X \subseteq S \cup F$.

Initial call: $\textsc{EnumCVC}(\emptyset, V(G))$.

Stop when: S is a minimal connected vertex cover or F is empty.

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v:

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F:

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$
- **discard** v:

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$
- **discard** v: delete v from F, add all free neighbors of v to S and delete them from F:

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$
- **discard** v: delete v from F, add all free neighbors of v to S and delete them from F: $\textsc{EnumCVC}(S \cup N(v), F \setminus N[v])$

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$
- **discard** v: delete v from F, add all free neighbors of v to S and delete them from F: $\textsc{EnumCVC}(S \cup N(v), F \setminus N[v])$

$T(n) \leq T(n-1) + T(n-2)$

Sketch of the proof

At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$
- **discard** v: delete v from F, add all free neighbors of v to S and delete them from F: $\textsc{EnumCVC}(S \cup N(v), F \setminus N[v])$

$T(n) \leq T(n-1) + T(n-2) \rightarrow T(n) = O(1.6181^n)$

Sketch of the proof

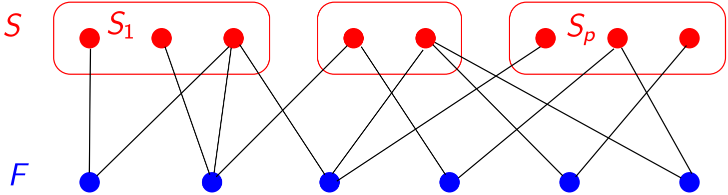
At every step, we pick a vertex and we branch on the possibilities of **selecting** it to be placed in S and **discarding** it from being placed in minimal connected vertex covers containing S.

If there are two adjacent vertices v and w in F then:

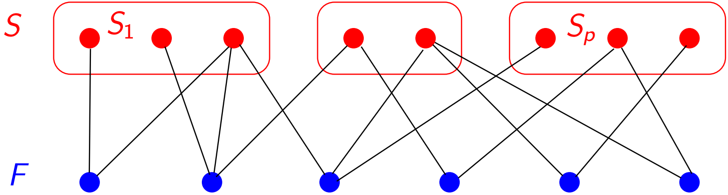
- **select** v: add v to S and delete v from F: $\textsc{EnumCVC}(S \cup \{v\}, F \setminus \{v\})$
- **discard** v: delete v from F, add all free neighbors of v to S and delete them from F: $\textsc{EnumCVC}(S \cup N(v), F \setminus N[v])$

$T(n) \leq T(n-1) + T(n-2) \rightarrow T(n) = O(1.6181^{n-|F|})$

Sketch of the proof

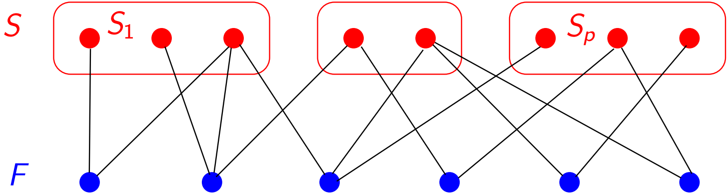


Sketch of the proof



If F is an independent set then check every subset of F of size at most $p - 1$, and combine it with S to see if it gives a minimal connected vertex cover.

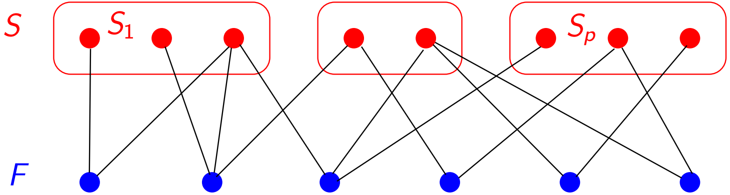
Sketch of the proof



If F is an independent set then check every subset of F of size at most $p-1$, and combine it with S to see if it gives a minimal connected vertex cover.

$$T(n) = O(1.6181^{n-|F|} \cdot 2^{|F|})$$

Sketch of the proof



If F is an independent set then check every subset of F of size at most $p-1$, and combine it with S to see if it gives a minimal connected vertex cover.

$$T(n) = O(1.6181^{n-|F|} \cdot 2^{|F|}) = O(1.8668^n)$$

(Balancing at $|F| = 2n/3$)

Graphs of chordality at most 5

The chordality of a graph G is the length of a longest induced (chordless) cycle in G.

Graphs of chordality at most 5

The chordality of a graph G is the length of a longest induced (chordless) cycle in G.

Theorem

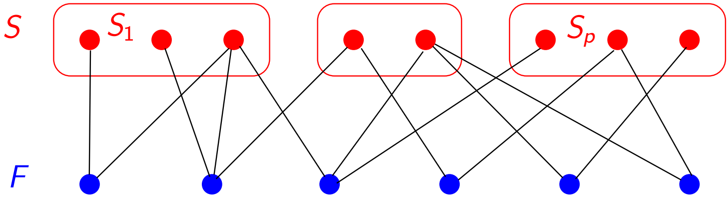
The maximum number of minimal connected vertex covers of a graph of chordality at most 5 is at most 1.6181^n, and these can be enumerated in time $O(1.6181^n)$.

Sketch of the proof

The algorithm starts exactly as the previous algorithm until F becomes an independent set:

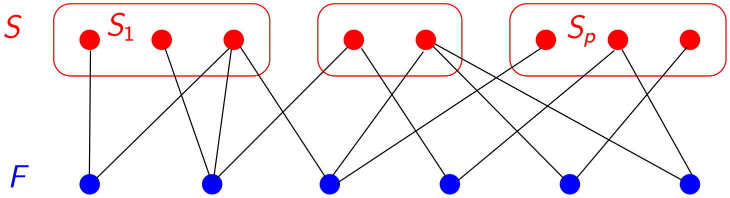
Sketch of the proof

The algorithm starts exactly as the previous algorithm until F becomes an independent set:



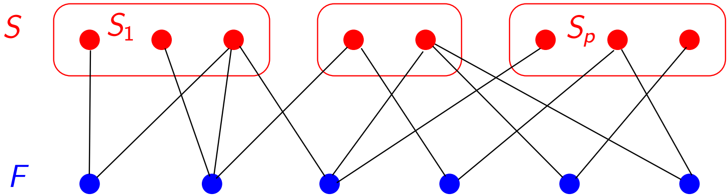
Sketch of the proof

The algorithm starts exactly as the previous algorithm until F becomes an independent set:

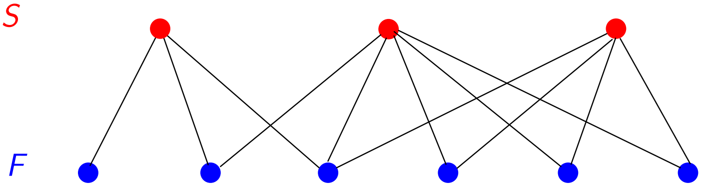


Recall that the running time of this part is dominated by $O(1.6181^n)$.

Sketch of the proof



Sketch of the proof



Sketch of the proof

Observations:

- All edges with both endpoints in S are already covered.

Sketch of the proof

Observations:

- All edges with both endpoints in S are already covered.
- Chordality does not increase when we contract edges.

Sketch of the proof

Observations:

- All edges with both endpoints in S are already covered.
- Chordality does not increase when we contract edges.
- We get a bipartite graph, hence of chordality at most 4.

Sketch of the proof

Observations:

- All edges with both endpoints in S are already covered.
- Chordality does not increase when we contract edges.
- We get a bipartite graph, hence of chordality at most 4. These are exactly the chordal bipartite graphs.

Sketch of the proof

Observations:

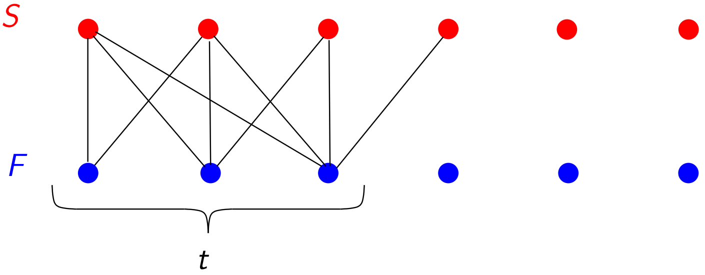
- All edges with both endpoints in S are already covered.
- Chordality does not increase when we contract edges.
- We get a bipartite graph, hence of chordality at most 4. These are exactly the chordal bipartite graphs.
- Every chordal bipartite graph has a weakly simplicial vertex in each partite set.

Sketch of the proof

A vertex in a graph is weakly simplicial if its neighborhood is an independent set and the neighborhoods of its neighbors form a chain under inclusion.

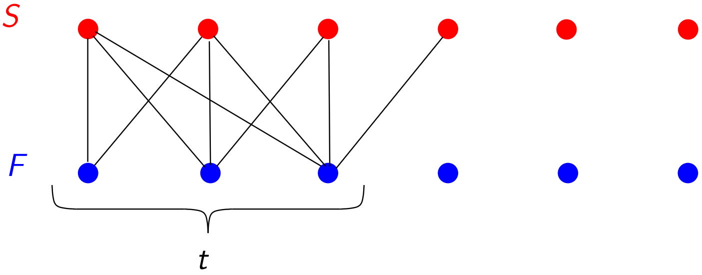
Sketch of the proof

A vertex in a graph is weakly simplicial if its neighborhood is an independent set and the neighborhoods of its neighbors form a chain under inclusion.



Sketch of the proof

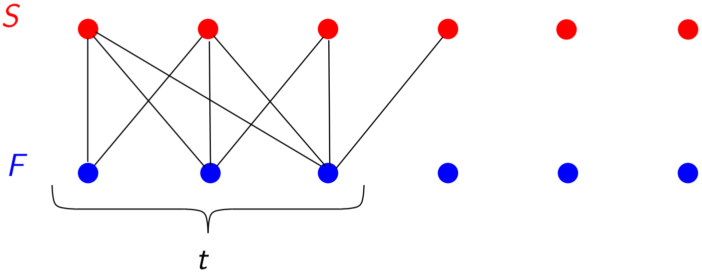
A vertex in a graph is weakly simplicial if its neighborhood is an independent set and the neighborhoods of its neighbors form a chain under inclusion.



For a weakly simplicial vertex outside F, we branch on its neighbors.

Sketch of the proof

A vertex in a graph is weakly simplicial if its neighborhood is an independent set and the neighborhoods of its neighbors form a chain under inclusion.



For a weakly simplicial vertex outside F, we branch on its neighbors. **Exactly one such neighbor can be selected.**

Sketch of the proof

Running time and maximum number of leaves:

$T(n) \leq t \cdot T(n - t)$

Sketch of the proof

Running time and maximum number of leaves:

$$T(n) \leq t \cdot T(n-t) \rightarrow T(n) = O(1.4423^n)$$

Sketch of the proof

Running time and maximum number of leaves:

$T(n) \leq t \cdot T(n-t) \rightarrow T(n) = O(1.4423^n)$

The running time and the maximum number of leaves of the search tree are dominated by the branching rule of the first part: $O(1.6181^n)$

Sketch of the proof

Running time and maximum number of leaves:

$T(n) \leq t \cdot T(n-t) \rightarrow T(n) = O(1.4423^n)$

The running time and the maximum number of leaves of the search tree are dominated by the branching rule of the first part: $O(1.6181^n)$

Observe that the lower bound example has chordality at most 5.

Chordal graphs

A graph is chordal if it has chordality at most 3.

Chordal graphs

A graph is chordal if it has chordality at most 3.

Theorem

The maximum number of minimal connected vertex covers of a chordal graph is at most $3^{n/3} < 1.4423^n$*, and these can be enumerated in time* $O(1.4423^n)$*.*

Chordal graphs

Lemma

Let G be a connected chordal graph and let C be its set of cut vertices.

Chordal graphs

Lemma

Let G be a connected chordal graph and let C be its set of cut vertices. Let $G' = G - C$.

Chordal graphs

Lemma

Let G be a connected chordal graph and let C be its set of cut vertices. Let $G' = G - C$.

S is a minimal vertex cover of G' if and only if $X = C \cup S$ is a minimal connected vertex cover of G.

Chordal graphs

Lemma

Let G be a connected chordal graph and let C be its set of cut vertices. Let $G' = G - C$.

S is a minimal vertex cover of G' if and only if $X = C \cup S$ is a minimal connected vertex cover of G.

Consequence: Enumerating minimal connected vertex covers is equivalent to enumerating minimal vertex covers,

Chordal graphs

Lemma

Let G be a connected chordal graph and let C be its set of cut vertices. Let $G' = G - C$.

S is a minimal vertex cover of G' if and only if $X = C \cup S$ is a minimal connected vertex cover of G.

Consequence: Enumerating minimal connected vertex covers is equivalent to enumerating minimal vertex covers, which are at most $3^{n/3} < 1.4423^n$ in number and can be enumerated in time $O(1.4423^n)$.

Sketch of the proof of lemma

If $X = C \cup S$ is a minimal connected vertex cover of G then S is a minimal vertex cover of G'

Sketch of the proof of lemma

If $X = C \cup S$ is a minimal connected vertex cover of G then S is a minimal vertex cover of G' since exactly the edges incident to vertices of C are covered by the vertices of C.

Sketch of the proof of lemma

If $X = C \cup S$ is a minimal connected vertex cover of G then S is a minimal vertex cover of G' since exactly the edges incident to vertices of C are covered by the vertices of C.

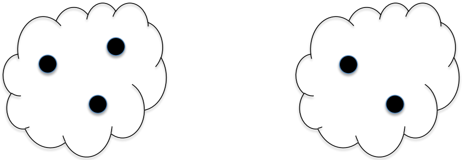
If S is a minimal vertex cover of G' then $X = C \cup S$ is a vertex cover of G.

Sketch of the proof of lemma

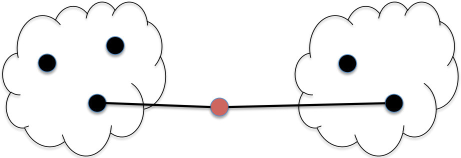
If $X = C \cup S$ is a minimal connected vertex cover of G then S is a minimal vertex cover of G' since exactly the edges incident to vertices of C are covered by the vertices of C.

If S is a minimal vertex cover of G' then $X = C \cup S$ is a vertex cover of G. We need to show that it is connected.

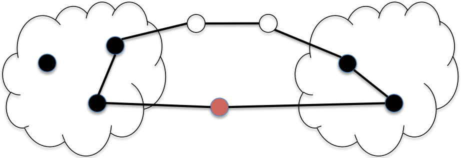
Sketch of the proof of lemma



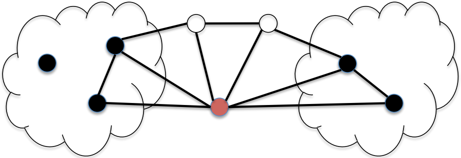
Sketch of the proof of lemma



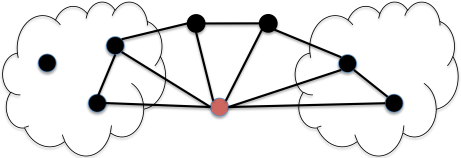
Sketch of the proof of lemma



Sketch of the proof of lemma



Sketch of the proof of lemma



Distance hereditary graphs

A graph G is distance-hereditary if for every connected induced subgraph H of G, $\mathrm{dist}_H(u, v) = \mathrm{dist}_G(u, v)$ for $u, v \in V(H)$.

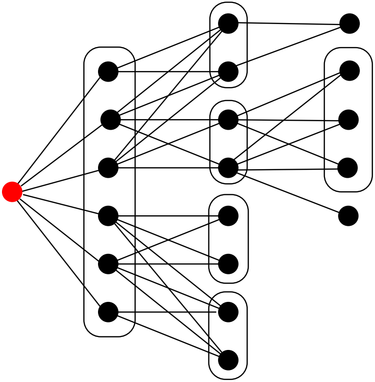
Distance hereditary graphs

A graph G is distance-hereditary if for every connected induced subgraph H of G, $\mathrm{dist}_H(u, v) = \mathrm{dist}_G(u, v)$ for $u, v \in V(H)$.

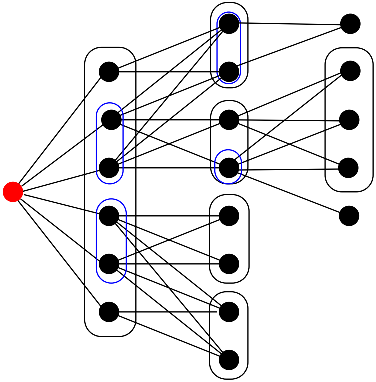
Theorem

The maximum number of minimal connected vertex covers of a distance-hereditary graph is at most $2 \cdot 3^{n/3}$, and these can be enumerated in time $O(1.4423^n)$.

Sketch of the proof



Sketch of the proof



Split graphs and cobipartite graphs

Split graphs and cobipartite graphs

Proposition

The number of minimal connected vertex covers of a split graph G is at most n, and these can be enumerated in time $O(n)$.

Split graphs and cobipartite graphs

Proposition

The number of minimal connected vertex covers of a split graph G is at most n, and these can be enumerated in time $O(n)$.

The number of minimal connected vertex covers of a cobipartite graph G is at most $n^2/4 + n$, and these can be enumerated in time $O(n^2)$.

Summary

Graph Class	Lower Bound	Upper Bound
general	$3^{(n-2)/3}$ $3^{(n-1)/3}$	2^n $O(1.8668^n)$
chordality≤ 5	$3^{(n-2)/3}$ $3^{(n-1)/3}$	2^n $O(1.6181^n)$
chordal	$3^{(n-2)/3}$ $3^{(n-1)/3}$	$O(1.7159^n)$ $3^{n/3}$
strongly chordal	$3^{(n-2)/3}$ $3^{(n-1)/3}$	$3^{n/3}$ $3^{n/3}$
split	1.3195^n n	1.3803^n n
cobipartite	1.3195^n $n^2/4$	1.3803^n $n^2/4+n$
interval	$3^{(n-2)/3}$ $3^{(n-1)/3}$	$3^{(n-2)/3}$ $3^{n/3}$
AT-free	$3^{(n-2)/3}$ $3^{(n-1)/3}$	$O^*(3^{n/3})$ $O(1.6181^n)$
distance-hereditary	$3^{(n-2)/3}$ $3^{(n-1)/3}$	$3^{n/3}\cdot n$ $2\cdot 3^{n/3}$
cographs	m $3^{(n-1)/3}$	m $2\cdot 3^{n/3}$

Minimal connected dominating sets and

minimal connected vertex covers

Open Questions : minimal connected vertex cover

- What is the maximum number of minimal connected vertex covers that a graph can have?

Open Questions : minimal connected vertex cover

- What is the maximum number of minimal connected vertex covers that a graph can have?
- We conjecture that it is more than $3^{n/3}$.

Open Questions : minimal connected vertex cover

- What is the maximum number of minimal connected vertex covers that a graph can have?
- We conjecture that it is more than $3^{n/3}$.
- How about graphs of bounded chordality?

Open questions : minimal connected dominating set

- Is it possible to enumerate all minimal connected dominating sets of a graph in time $O(c^n)$ for $c < 2$?

Open questions : minimal connected dominating set

- Is it possible to enumerate all minimal connected dominating sets of a graph in time $O(c^n)$ for $c < 2$?
- Is it possible to get a non-algorithmic upper bound for the number of minimal connected dominating sets?

Open questions : minimal connected dominating set

- Is it possible to enumerate all minimal connected dominating sets of a graph in time $O(c^n)$ for $c < 2$?
- Is it possible to get a non-algorithmic upper bound for the number of minimal connected dominating sets?
- Is it possible to improve the lower bound $3^{(n-2)/3}$?

Open questions : minimal connected dominating set

- Is it possible to enumerate all minimal connected dominating sets of a graph in time $O(c^n)$ for $c < 2$?
- Is it possible to get a non-algorithmic upper bound for the number of minimal connected dominating sets?
- Is it possible to improve the lower bound $3^{(n-2)/3}$?
- Is it possible to improve the upper bound $O(1.3803^n)$ for split graphs?

Thank You!