A new algorithmic framework for enumerating commutable set properties

Luca Versari

July 21, 2017

Set systems

Definitions

Definition
A **set system** $\mathcal{F}$ over a ground set E is a family of subsets of E, i.e. $\mathcal{F} \subseteq 2^E$.

Definitions

Definition
A **set system** $\mathcal{F}$ over a ground set E is a family of subsets of E, i.e. $\mathcal{F} \subseteq 2^E$.

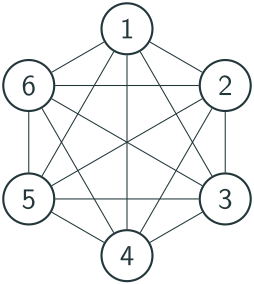
Definition
An **independence system** is a set system that is closed under subsets, so that if $A \in \mathcal{F}$ then any $B \subseteq A$ also belongs to $\mathcal{F}$.

Example: clique

Definition
A **clique** in a graph G is a subset of the nodes such that any two of them are connected by an edge.

Example: clique

Definition
A **clique** in a graph G is a subset of the nodes such that any two of them are connected by an edge.



Definitions

Definition
A **strongly accessible set system** is a set system that satisfies the following property: given $A, B \in \mathcal{F}$, if $A \subsetneq B$ then there is an $a \in B \setminus A$ such that $A \cup \{a\} \in \mathcal{F}$.

Definition
A **strongly accessible set system** is a set system that satisfies the following property: given $A, B \in \mathcal{F}$, if $A \subsetneq B$ then there is an $a \in B \setminus A$ such that $A \cup \{a\} \in \mathcal{F}$.

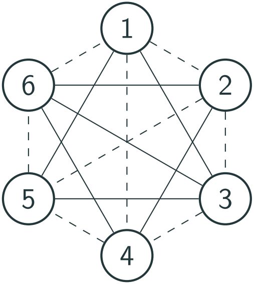
Definition
A **commutable set system** is a strongly accessible set system that satisfies the following property: given $A, B, C, D \in \mathcal{F}$, if $A \subset B \cap C$, $B \cup C \subset D$ then $B \cup C \in \mathcal{F}$.

Example: black-connected cliques

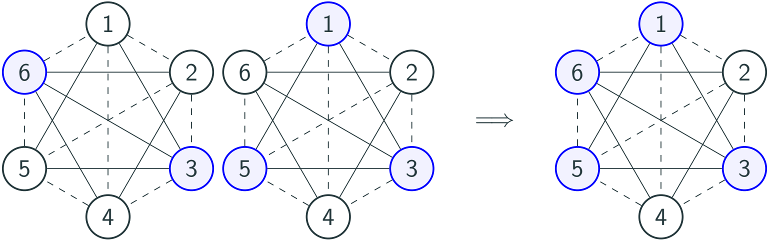
Definition
In a graph with both black and white edges, a **black-connected clique** is a subset of the nodes such that it is a clique and it is connected while considering only black edges.

Example: black-connected cliques

Definition
In a graph with both black and white edges, a **black-connected clique** is a subset of the nodes such that it is a clique and it is connected while considering only black edges.



The “commutable” property for black-connected cliques

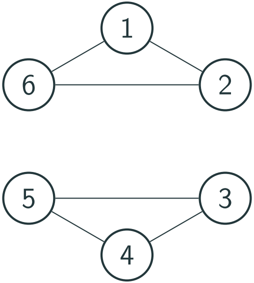


Example: k-plex

Definition
A k-**plex** in a graph G is a subset of the nodes such that for any node v there are at most k nodes that do not have an edge with v.

Example: k-plex

Definition
A k-**plex** in a graph G is a subset of the nodes such that for any node v there are at most k nodes that do not have an edge with v.



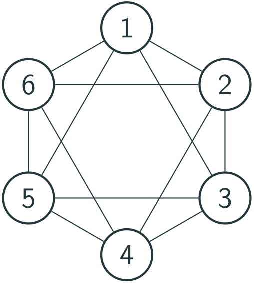
$k = 4$

Example: connected k-plex

Definition
A **connected k-plex** in a graph G is a subset of the nodes that is both a k-plex and connected.

Example: connected k-plex

Definition
A **connected k-plex** in a graph G is a subset of the nodes that is both a k-plex and connected.



$k = 2$

Our goal

The objective of our framework is to **enumerate all the maximal sets** (by inclusion) in a commutable set system.

Reverse search

Definition
Let $\mathcal{F}$ be a strongly accessible set system. We say that *parent* is a **parent function** if it has the following properties:

- It is defined on all the maximal sets in $\mathcal{F}$, except for some of them we will call the *root*s.

Definition
Let $\mathcal{F}$ be a strongly accessible set system. We say that *parent* is a **parent function** if it has the following properties:

- It is defined on all the maximal sets in $\mathcal{F}$, except for some of them we will call the *root*s.
- *parent*(S) is another maximal set in $\mathcal{F}$, for all the maximal solutions S for which it is defined (so for all the S that are not a *root*).

Definition
Let $\mathcal{F}$ be a strongly accessible set system. We say that *parent* is a **parent function** if it has the following properties:

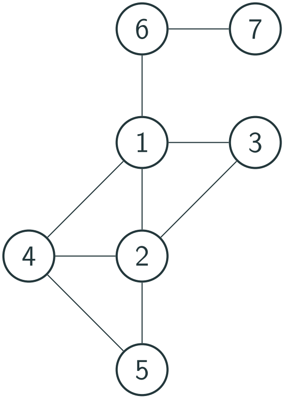
- It is defined on all the maximal sets in $\mathcal{F}$, except for some of them we will call the *root*s.
- $\mathit{parent}(S)$ is another maximal set in $\mathcal{F}$, for all the maximal solutions S for which it is defined (so for all the S that are not a *root*).
- There is an order $\prec$ such that $\mathit{parent}(S) \prec S$ for all the S that are not a *root*.

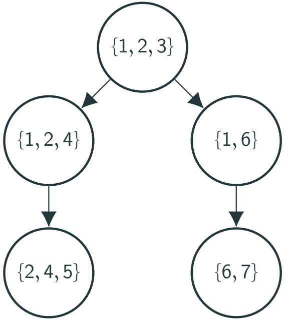
Definition
If *parent* is a parent function, we define $\mathit{children}(S)$ for a maximal solution S to be the set of all maximal solutions C such that $\mathit{parent}(C) = S$.

Definition
If *parent* is a parent function, we define $\mathit{children}(S)$ for a maximal solution S to be the set of all maximal solutions C such that $\mathit{parent}(C) = S$.

Theorem
Let G be the directed graph that has the maximal solutions of $\mathcal{F}$ as nodes, and has an outgoing edge from S to any solution in $\mathit{children}(S)$. Then G is a directed forest rooted in the solutions that are roots.

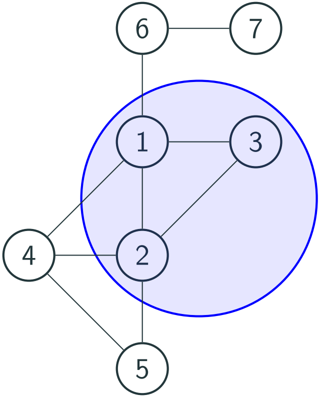
On the right is a possible reverse search graph for the cliques in the graph on the left.

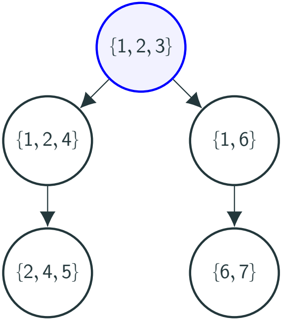




Example

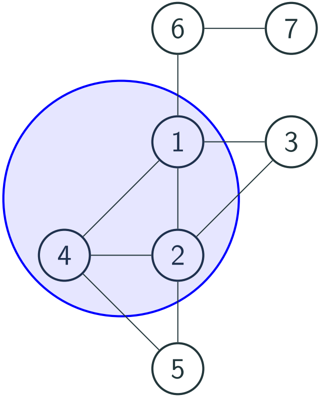
On the right is a possible reverse search graph for the cliques in the graph on the left.

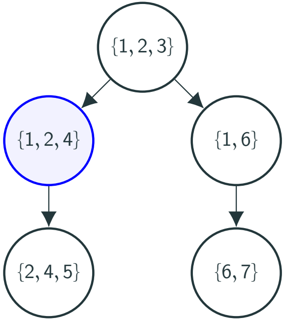




Example

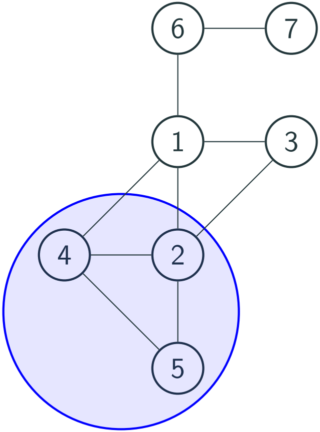
On the right is a possible reverse search graph for the cliques in the graph on the left.

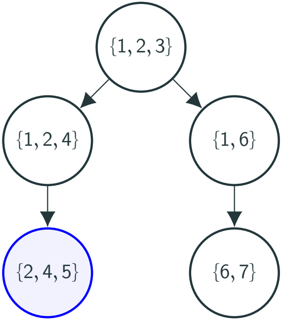




Example

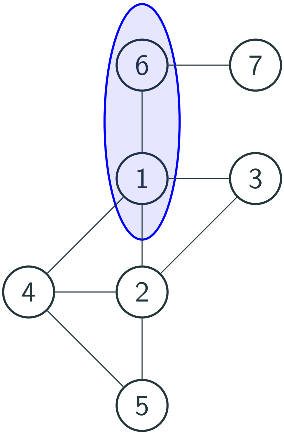
On the right is a possible reverse search graph for the cliques in the graph on the left.

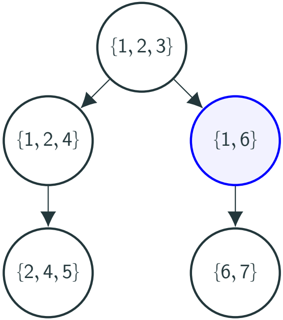




Example

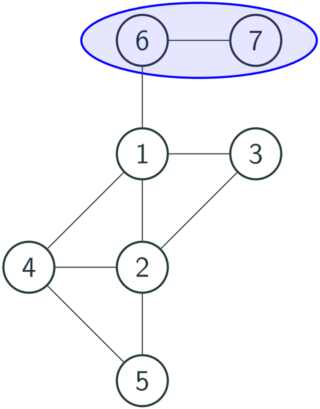
On the right is a possible reverse search graph for the cliques in the graph on the left.

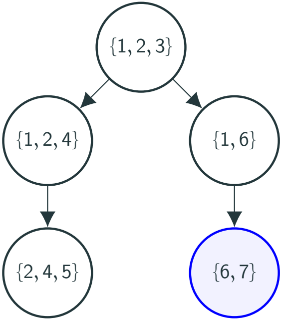




Example

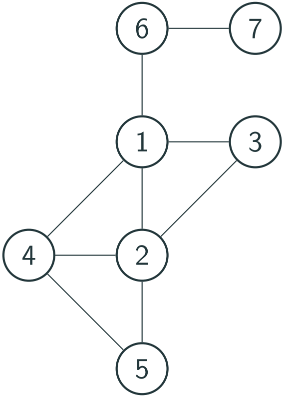
On the right is a possible reverse search graph for the cliques in the graph on the left.

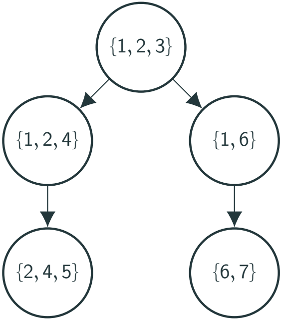




Example

On the right is a possible reverse search graph for the cliques in the graph on the left.

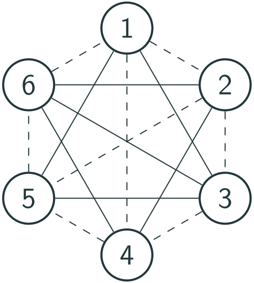




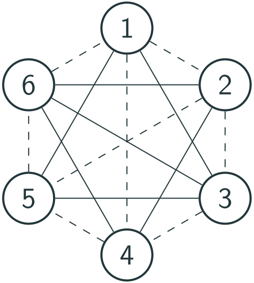
parent and *children* for commutable set systems

Definition
Given a commutable set system $\mathcal{F}$ and one of its feasible sets S, we say that s is a **seed** of S if $s \in S$ and $\{s\} \in \mathcal{F}$. The **canonical seed** of S, denoted by $seed(S)$, is the smallest possible seed according to the ordering of the elements of E.

Example of these definitions



Example of these definitions



$$seed(S) = 1$$

Definition
Given a commutable set system $\mathcal{F}$, one of its feasible sets S and a seed s of S, the **level** of an element v with respect to s $(level_S^s(v))$ is defined as follows:

- if $v = s$, then the level of v is 0.

Level

Definition
Given a commutable set system $\mathcal{F}$, one of its feasible sets S and a seed s of S, the **level** of an element v with respect to s ($\mathit{level}_S^s(v)$) is defined as follows:

- if $v = s$, then the level of v is 0.
- the level of v is $k + 1$ if k is the smallest integer such that there is a $S' \subseteq S$ that satisfies:

Definition
Given a commutable set system $\mathcal{F}$, one of its feasible sets S and a seed s of S, the **level** of an element v with respect to s ($level_S^s(v)$) is defined as follows:

- if $v = s$, then the level of v is 0.
- the level of v is $k + 1$ if k is the smallest integer such that there is a $S' \subseteq S$ that satisfies:
  - its elements have level $\leqslant k$

Level

Definition
Given a commutable set system $\mathcal{F}$, one of its feasible sets S and a seed s of S, the **level** of an element v with respect to s $\left(level_S^s(v)\right)$ is defined as follows:

- if $v = s$, then the level of v is 0.
- the level of v is $k + 1$ if k is the smallest integer such that there is a $S' \subseteq S$ that satisfies:
  - its elements have level $\leqslant k$
  - $S' \cup \{v\} \in \mathcal{F}$

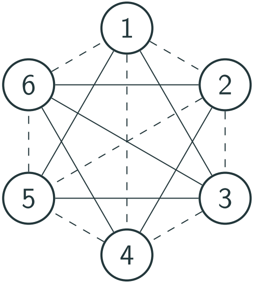
Definition
Given a commutable set system $\mathcal{F}$, one of its feasible sets S and a seed s of S, the **level** of an element v with respect to s ($level_S^s(v)$) is defined as follows:

- if $v = s$, then the level of v is 0.
- the level of v is $k+1$ if k is the smallest integer such that there is a $S' \subseteq S$ that satisfies:
  - its elements have level $\leqslant k$
  - $S' \cup \{v\} \in \mathcal{F}$
  - $s \in S'$.

Definition
Given a commutable set system $\mathcal{F}$, one of its feasible sets S and a seed s of S, the **level** of an element v with respect to s ($level_S^s(v)$) is defined as follows:

- if $v = s$, then the level of v is 0.
- the level of v is $k + 1$ if k is the smallest integer such that there is a $S' \subseteq S$ that satisfies:
  - its elements have level $\leqslant k$
  - $S' \cup \{v\} \in \mathcal{F}$
  - $s \in S'$.
- if there is no such subset, we say that the level of v is ∞.

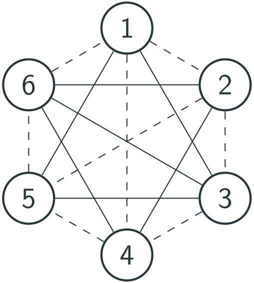
Example of these definitions

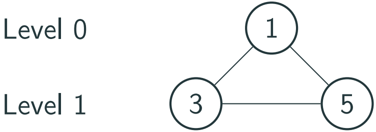


Level 0

$$seed(S) = 1$$

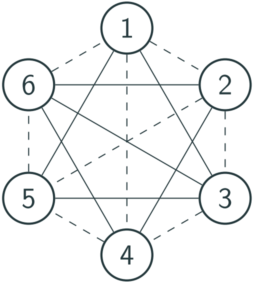
Example of these definitions

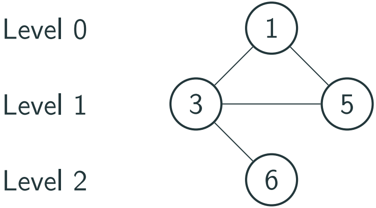




$$seed(S) = 1$$

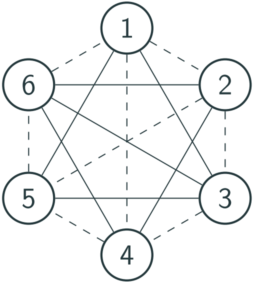
Example of these definitions

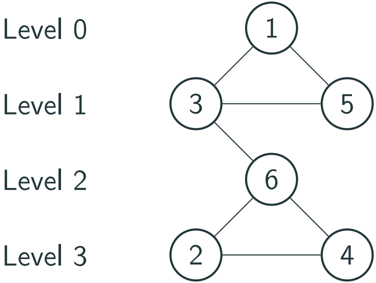




$$seed(S) = 1$$

Example of these definitions





$$seed(S) = 1$$

Level order

Definition
The **level order** $\prec$ between any two solutions P, Q of a commutable set system $\mathcal{F}$ is defined as follows:

- Let $lP = [(level_P(v), v) \ \forall v \in P]$, the tuple of pairs made by the level of an element and the element itself, sorted in increasing order.

Level order

Definition

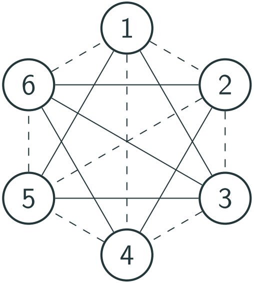
The **level order** $\prec$ between any two solutions P, Q of a commutable set system $\mathcal{F}$ is defined as follows:

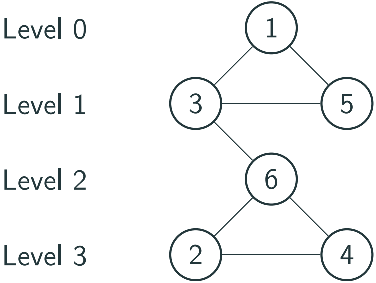
- Let $lP = [(level_P(v), v)\ \forall v \in P]$, the tuple of pairs made by the level of an element and the element itself, sorted in increasing order.
- Let lQ be defined in the same way.

Definition
The **level order** $\prec$ between any two solutions P, Q of a commutable set system $\mathcal{F}$ is defined as follows:

- Let $lP = [(level_P(v), v) \ \forall v \in P]$, the tuple of pairs made by the level of an element and the element itself, sorted in increasing order.
- Let lQ be defined in the same way.
- We say that $P \prec Q$ if and only if lP is smaller than lQ according to lexicographical order.

Example of these definitions





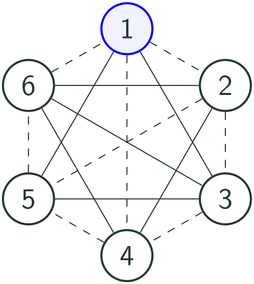
$$seed(S) = 1$$

$$IS = [(0,1),(1,3),(1,5),(2,6),(3,2),(3,4)]$$

Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

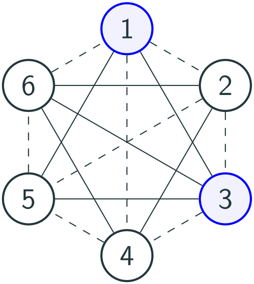
Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$complete(\{1\}) = [(0, 1) \qquad]$$



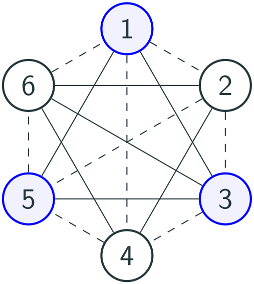
Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$\mathit{complete}(\{1\}) = [(0,1),(1,3) \qquad]$$



Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

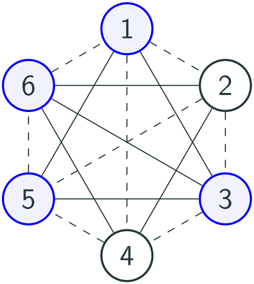
$$complete(\{1\}) = [(0, 1), (1, 3), (1, 5) \qquad]$$



Definition

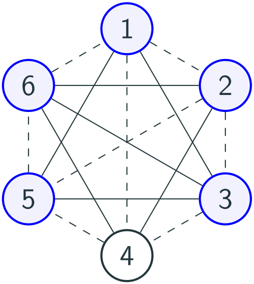
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$\textit{complete}(\{1\}) = [(0,1),(1,3),(1,5),(2,6) \qquad\qquad]$$



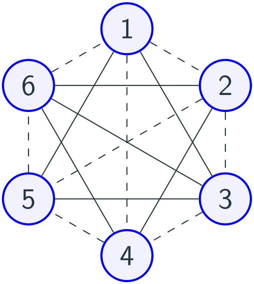
Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$complete(\{1\}) = [(0,1),(1,3),(1,5),(2,6),(3,2) \qquad]$$



Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$\textit{complete}(\{1\}) = [(0,1),(1,3),(1,5),(2,6),(3,2),(3,4)]$$



Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$\mathit{complete}(\{1\}) = [(0,1),(1,3),(1,5),(2,6),(3,2),(3,4)]$$

Definition
Given a maximal solution S, *parent*(S) is defined as *complete* of the longest prefix P of S such that *complete*$(P) \neq S$. This prefix is called *core*(S) and the next element in S according to level order is called *parind*(S).

Definition
Given a non-maximal solution S of a commutable set system $\mathcal{F}$, *complete*(S) is defined as the solution obtained by iteratively adding the smallest-level element to S. In case of ties, the node with the lowest label is chosen.

$$\textit{complete}(\{1\}) = [(0,1),(1,3),(1,5),(2,6),(3,2),(3,4)]$$

Definition
Given a maximal solution S, *parent*(S) is defined as *complete* of the longest prefix P of S such that *complete*$(P) \neq S$. This prefix is called *core*(S) and the next element in S according to level order is called *parind*(S).
We can prove that *parent*$(S) \prec S$.

Definition
Given a strongly accessible set system $\mathcal{F}$ on E, a maximal feasible set S and an element $v \in E \setminus S$, the **restricted problem** $\mathcal{P}(S, v)$ is the problem of enumerating all the maximal elements of the family

$$\mathcal{G}_S^v = \{A \in \mathcal{F} : A \subseteq S \cup \{v\}\}$$

Definition
Given a strongly accessible set system $\mathcal{F}$ on E, a maximal feasible set S and an element $v \in E \setminus S$, the **restricted problem** $\mathcal{P}(S, v)$ is the problem of enumerating all the maximal elements of the family

$$\mathcal{G}_S^v = \{A \in \mathcal{F} : A \subseteq S \cup \{v\}\}$$

Theorem
Given a maximal solution C such that $parent(C) = S$, $core(C)$ is the prefix ending just before $parind(C)$ of a solution of $\mathcal{P}(S, parind(C))$.

Applications

Cliques and black-connected cliques

In both cases, the restricted problem is easy and has at most one solution. Let q be the maximum size of a black-connected clique. Then

Theorem
All the maximal black-connected cliques in a graph G may be enumerated in $O(q^5\Delta_b^2)$ time per solution, using only $O(q)$ extra memory (other than the memory used to store G).

Connected k-plexes

Lemma
Assuming k to be a constant, if S is a k-plex and v is a node in $V \setminus S$, then there are at most $1 + f(k)|S|^{k-1}$ maximal k-plexes in $S \cup \{v\}$, with $f(k) = (k-1)^{2k}$ for $k > 1$ and $f(1) = 1$. Moreover, they can be computed in $O(kf(k)|S|^k)$ time using only $O(kq)$ memory.

Connected k-plexes

Lemma
Assuming k to be a constant, if S is a k-plex and v is a node in $V \setminus S$, then there are at most $1 + f(k)|S|^{k-1}$ maximal k-plexes in $S \cup \{v\}$, with $f(k) = (k-1)^{2k}$ for $k > 1$ and $f(1) = 1$. Moreover, they can be computed in $O(kf(k)|S|^k)$ time using only $O(kq)$ memory.

Theorem
All the connected k-plexes in a graph G can be enumerated in $O(q^{k+4}\Delta^2 f(k))$ time per solution, using only $O(kq)$ extra memory (other than the memory used to store G).

Conclusions

- We have obtained a framework that achieves polynomial total time enumeration with low memory of suitable set families

Conclusions

- We have obtained a framework that achieves polynomial total time enumeration with low memory of suitable set families
- We have studied some families and shown how to apply the framework

Conclusions

- We have obtained a framework that achieves polynomial total time enumeration with low memory of suitable set families
- We have studied some families and shown how to apply the framework
- Future work: an experimental evaluation of the algorithms obtained

Conclusions

- We have obtained a framework that achieves polynomial total time enumeration with low memory of suitable set families
- We have studied some families and shown how to apply the framework
- Future work: an experimental evaluation of the algorithms obtained
- Future work: extending the framework to more general families

Any questions?