Fast-Mixed Searching and Related Problems on Graphs

Boting Yang
Department of Computer Science
University of Regina

Outline

- Fast searching and mixed searching
- Induced-path cover
- Fast-mixed searching vs fast searching
- Fast-mixed searching vs mixed searching
- Fast-mixed searching vs induced-path cover
- Characterizations
- Algorithms
- Complexity

Graph searching models

- Edge searching (Megiddo, Hakimi, Garey, Johnson and Papadimitriou, 1981)
- Node searching (Kirousis and Papadimitriou, 1986)
- Mixed searching (Bienstock and Seymour, 1991)
- Fast searching (Dyer, Yang, Yasar, 2008)

Mixed searching model

- At each step, only one of the following three actions is allowed:
  - placing a searcher on a vertex,
  - sliding a searcher along an edge,
  - removing a searcher from a vertex.

Three ways to clear an edge *uv* in mixed searching

- Edge *uv* becomes cleared if both endpoints are occupied by searchers
- sliding a searcher from *u* to *v* along *uv* while at least one searcher is located on *u*.
- sliding a searcher from *u* to *v* along *uv* while all edges incident on *u* except *uv* are already cleared.

Fast searching model

- At each step, only one of the following two actions is allowed:
  - placing a searcher on a vertex,
  - sliding a searcher along an edge.

- Each edge is traversed exactly once.

Two ways to clear an edge *uv* in fast searching

- sliding a searcher from *u* to *v* along *uv* while at least one searcher is located on *u*.

- sliding a searcher from *u* to *v* along *uv* while all edges incident on *u* except *uv* are already cleared.

What is fast-mixed searching?

- Fast-mixed searching is a combination of fast searching and mixed searching.
- A graph contains a fugitive hiding on vertices or along edges.
- The basic goal in a fast-mixed search is to use the minimum number of searchers to capture the fugitive.

Motivations

- In some real-life scenarios, the cost of a searcher may be relatively low in comparison to the cost of allowing a fugitive to be free for a long period of time.
- A fast-mixed search strategy of a graph gives an induced-path cover of the graph.
- Task scheduling

In fast-mixed searching, the fugitive …

- can stay on edges or on vertices.
- has complete knowledge of the location of every searcher.
- is invisible to searchers.
- can move in the graph along any path that does not include a searcher.
- always takes the best strategy for him to avoid being captured.

Some definitions...

- An edge the fugitive could be on is said to be contaminated.
- An edge the fugitive cannot be on is said to be cleared.

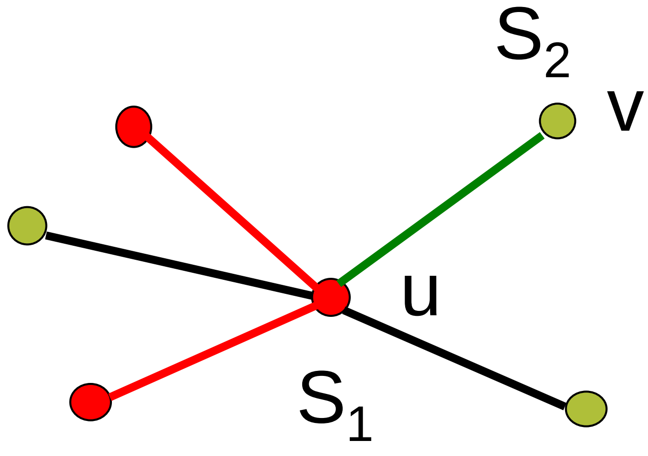
In fast-mixed searching, searchers …

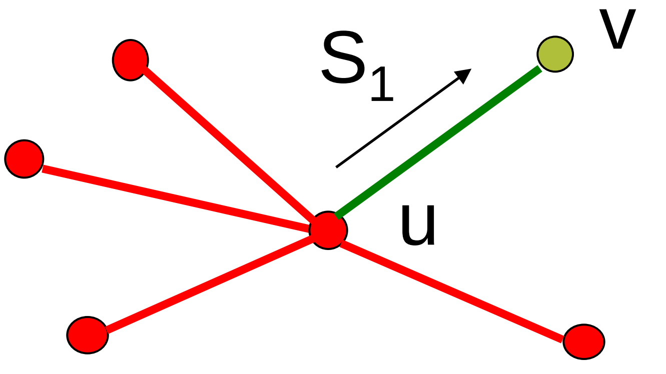
- have only one of the following two actions at each step:
  - placing a searcher on a contaminated vertex, or
  - sliding a searcher along a contaminated edge *uv* from *u* to *v* if *v* is contaminated and all edges incident on *u* except *uv* are cleared.

Two ways to clear an edge *uv* in fast-mixed searching:

- edge *uv* becomes cleared if both endpoints are occupied by searchers, or
- edge *uv* becomes cleared if a searcher slides along *uv* from *u* to *v* if *v* is contaminated and all edges incident on *u* except *uv* are cleared.

Two ways to clear an edge *uv* in fast-mixed searching:





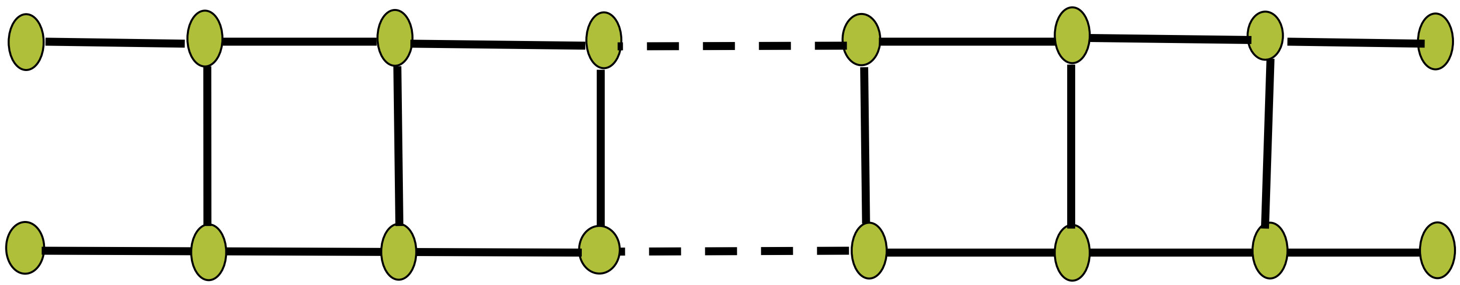
Fast-mixed searching strategy and number

- A fast-mixed search strategy (fms-strategy) of G is a sequence of actions such that the final action leaves all edges of G cleared.
- The minimum number of searchers required to clear G is the fast-mixed search number of G, denoted by fms(G).

Fast search number and mixed search number

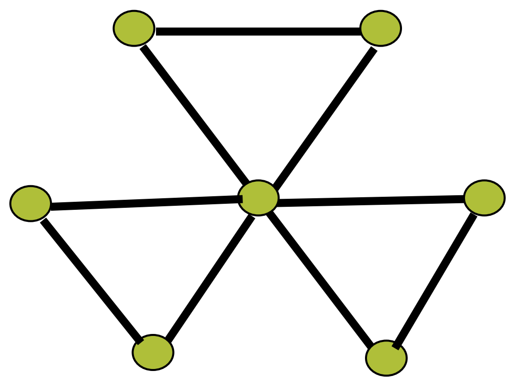
- The minimum number of searchers required to clear graph *G* in fast searching is the fast search number of *G*, denoted *fs(G)*.
- The minimum number of searchers required to clear graph *G* in mixed searching is the mixed search number of *G*, denoted *ms(G)*.

fs(G)/fms(G) can be arbitrarily large for standard ladders



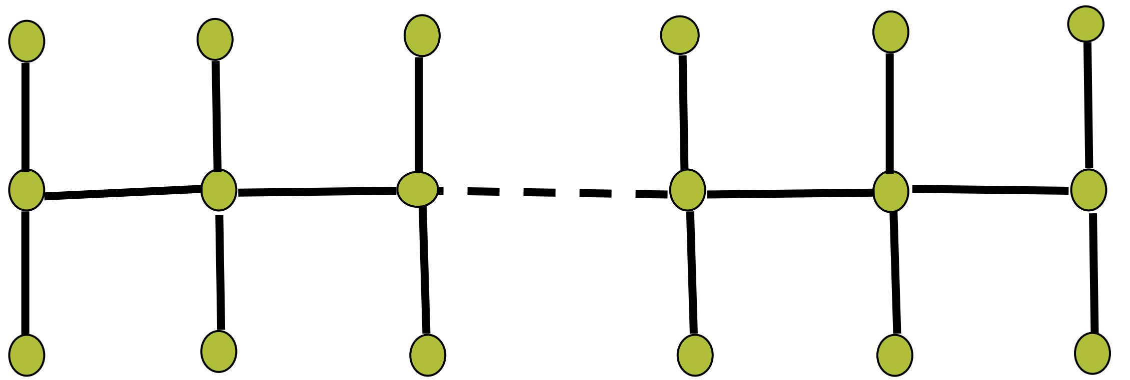
$$fs(L_k) = k + 2,\ fms(L_k) = 2$$

$fs(G)/fms(G)$ can be arbitrarily small positive number



$$fs(G_k) = 2,\ fms(G_k) = k + 1$$

fms(G)/ms(G) can be arbitrarily large for caterpillars



$$fms(H_k) = k, ms(H_k) = 2$$

Fast-mixed searching can be very different from fast searching and mixed searching

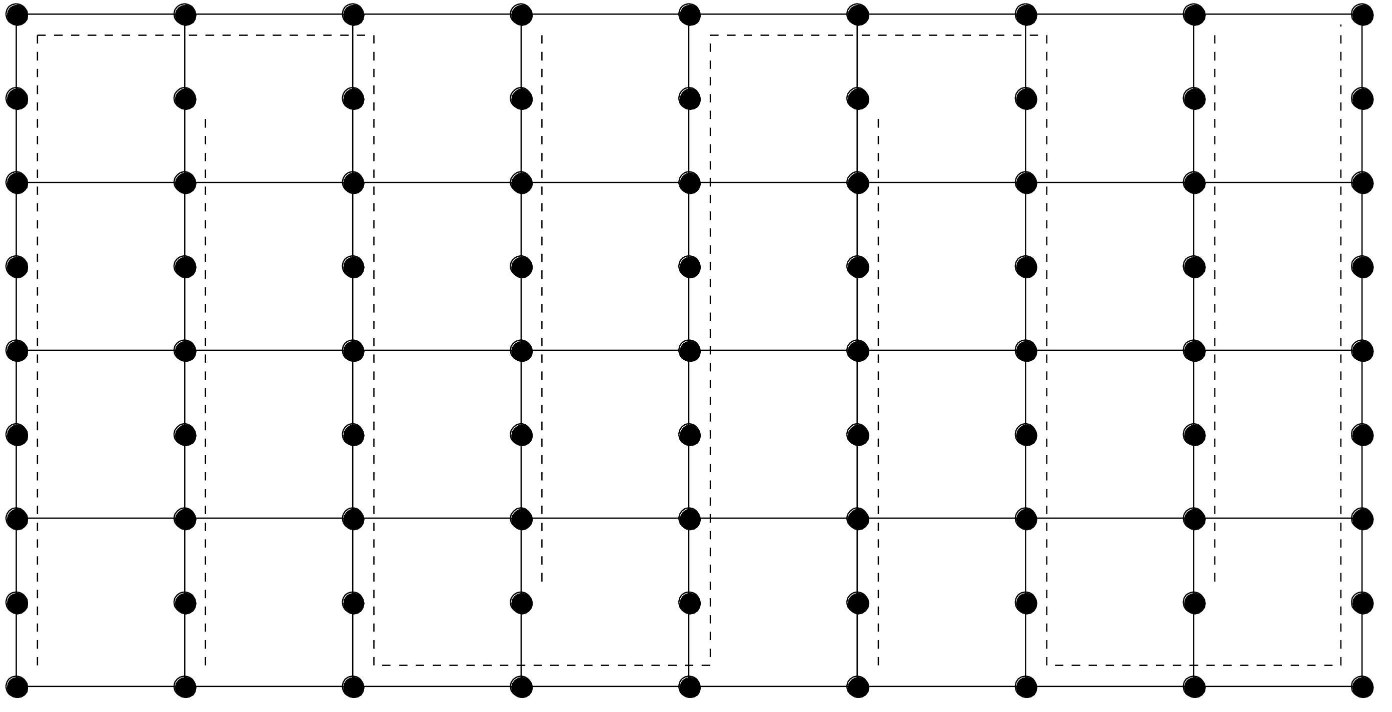
Theorem. Given a graph *G* that contains at least one edge, let *G'* be a graph obtained from *G* by adding two pendent edges on each vertex. Then

$$fs(G) \leq fs(G') \leq |V(G)| + fs(G),$$

$$ms(G') = ms(G) + 1,$$

and $fms(G') = ipc(G') = |V(G)|$.

Fast-mixed searching, fast searching, mixed searching, and induced-path cover



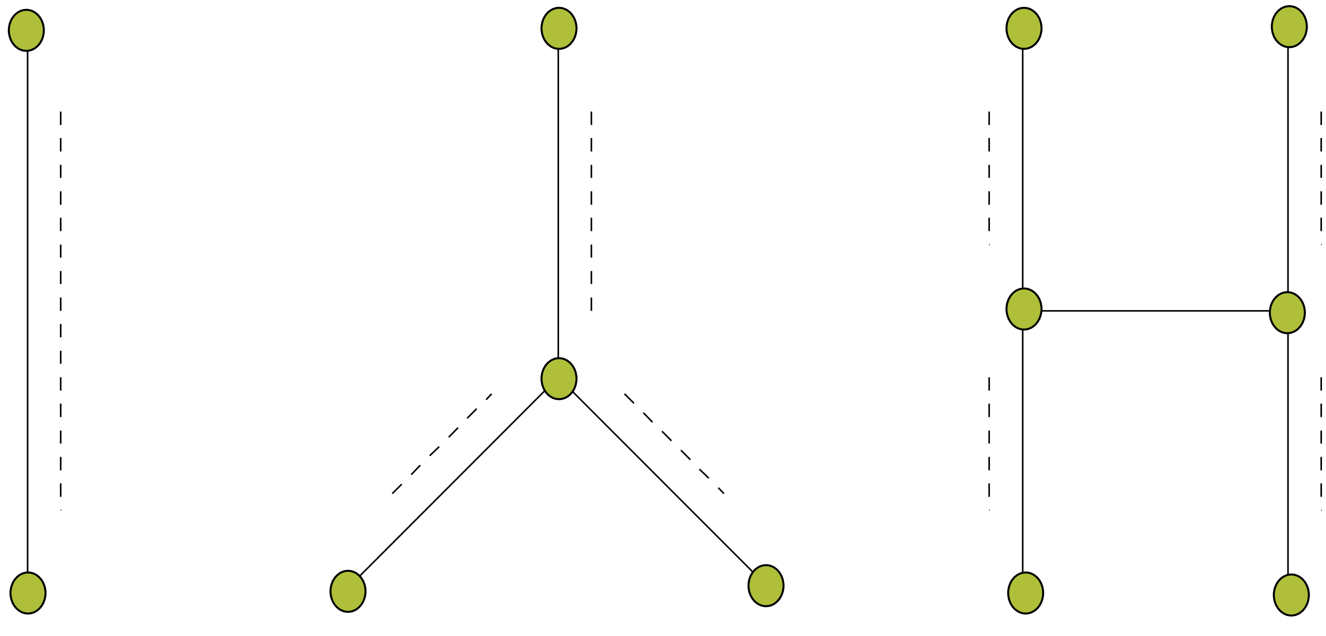
$$fms(G) = 9,\ fs(G) = 12, ms(G) = 6, ipc(G) \leq 5$$

Characterizations (I)

Theorem. For a tree T, the following are equivalent:

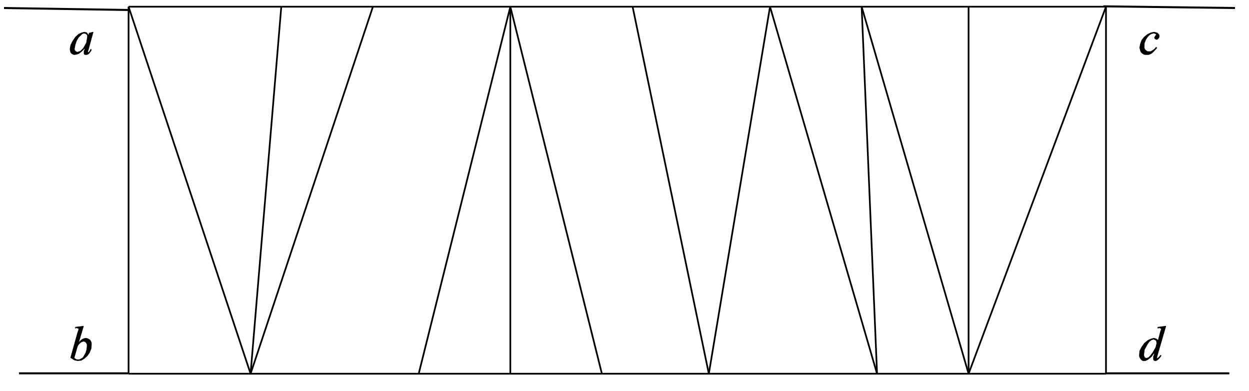
- fms(T) < 3.
- All vertices of T have degree at most 3; at most two vertices have degree 3; and if T has two vertices of degree 3, then these two vertices must be adjacent.
- T is one of the graphs in the following figure

Trees with fms ≤ 2

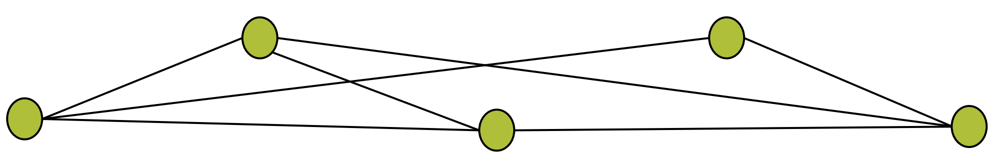


Characterizations (II)

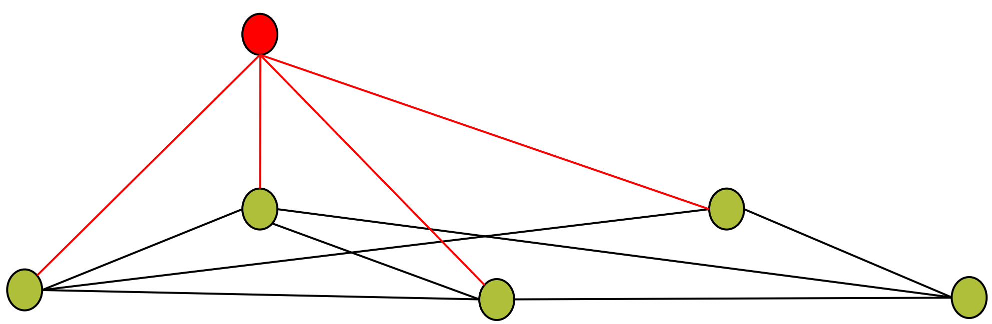
- Theorem. For any connected graph G that is not a tree, fms(G) = 2 if and only if G is a ladder.



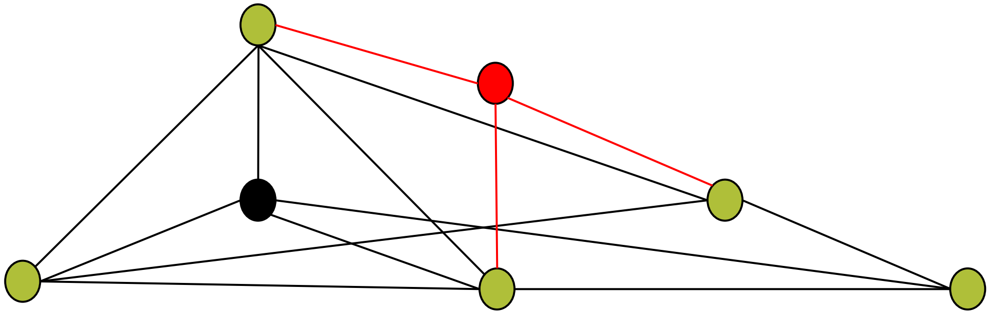
k-stack



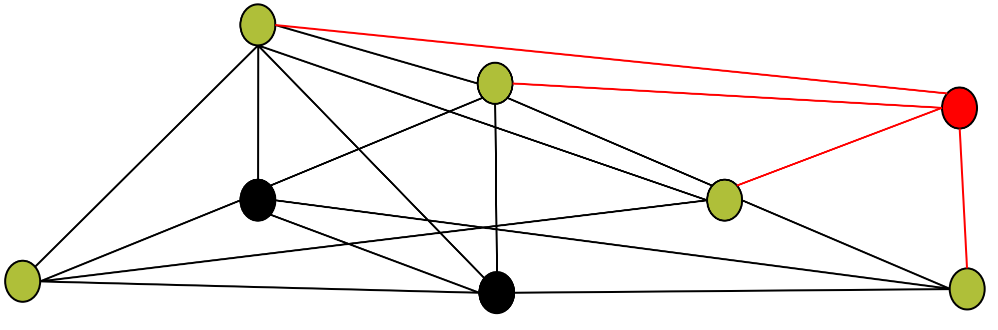
k-stack



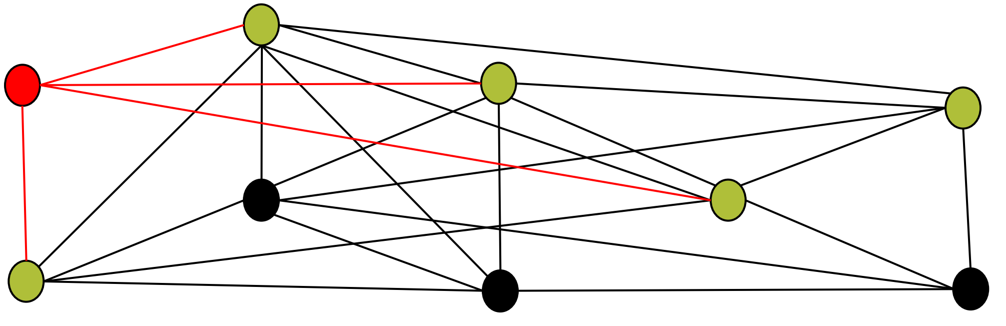
k-stack



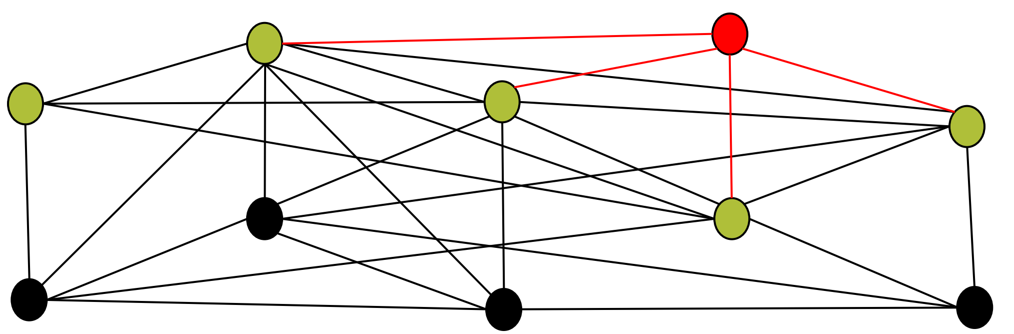
k-stack



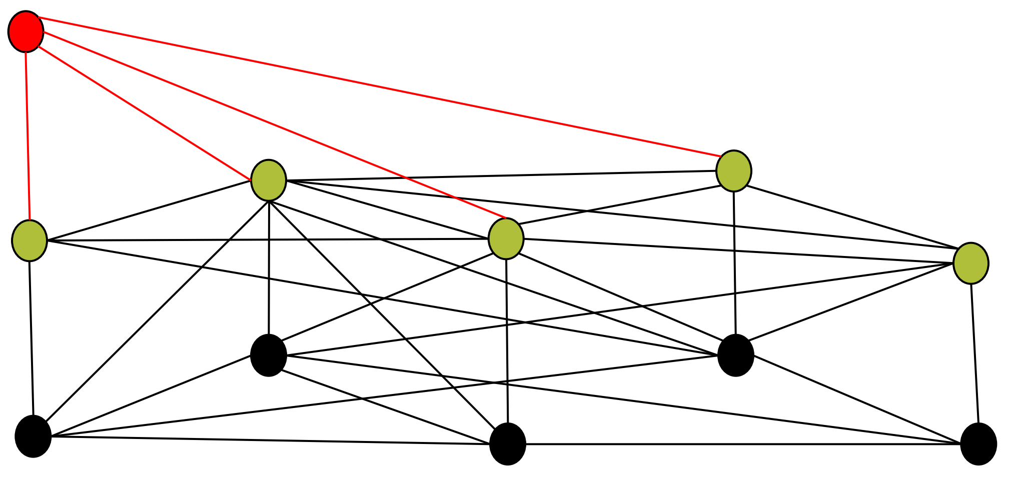
k-stack



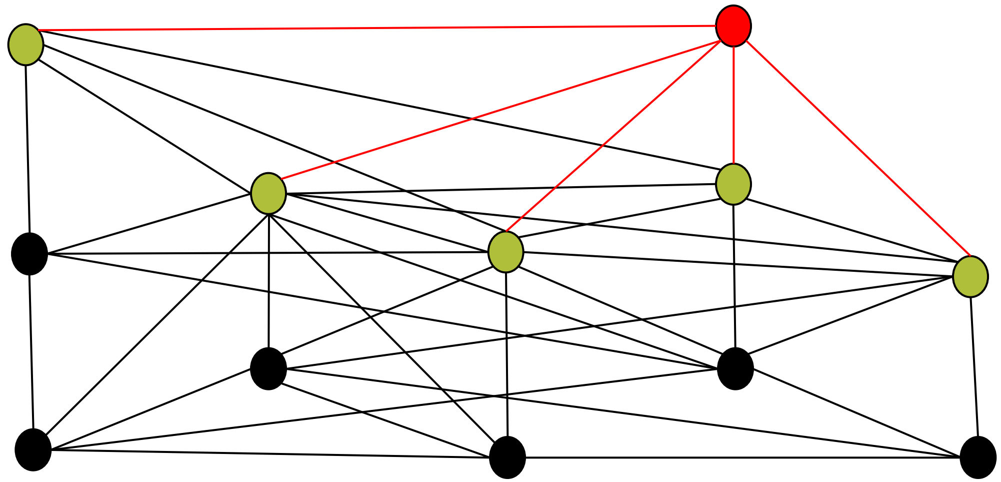
k-stack



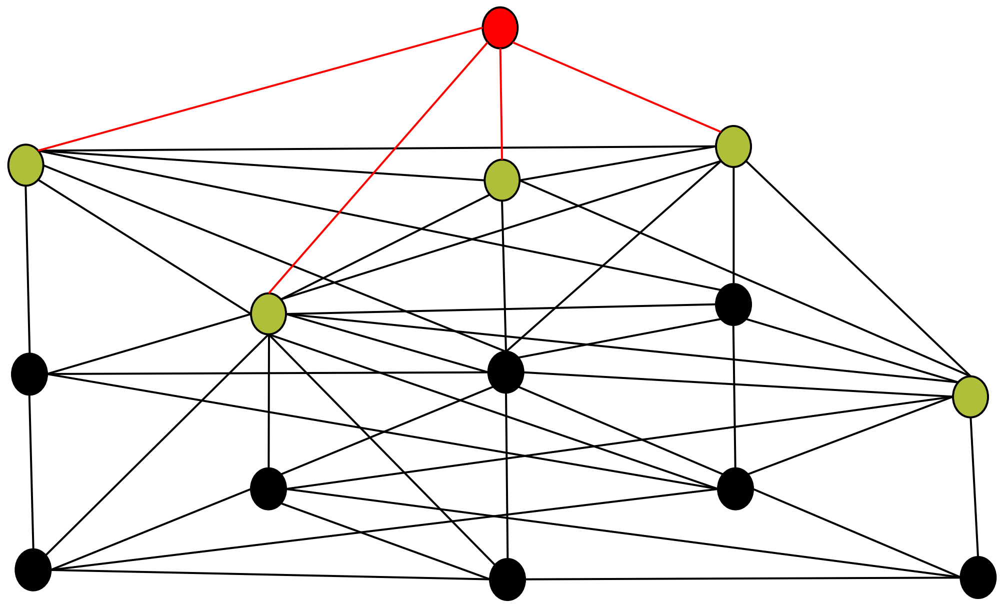
k-stack



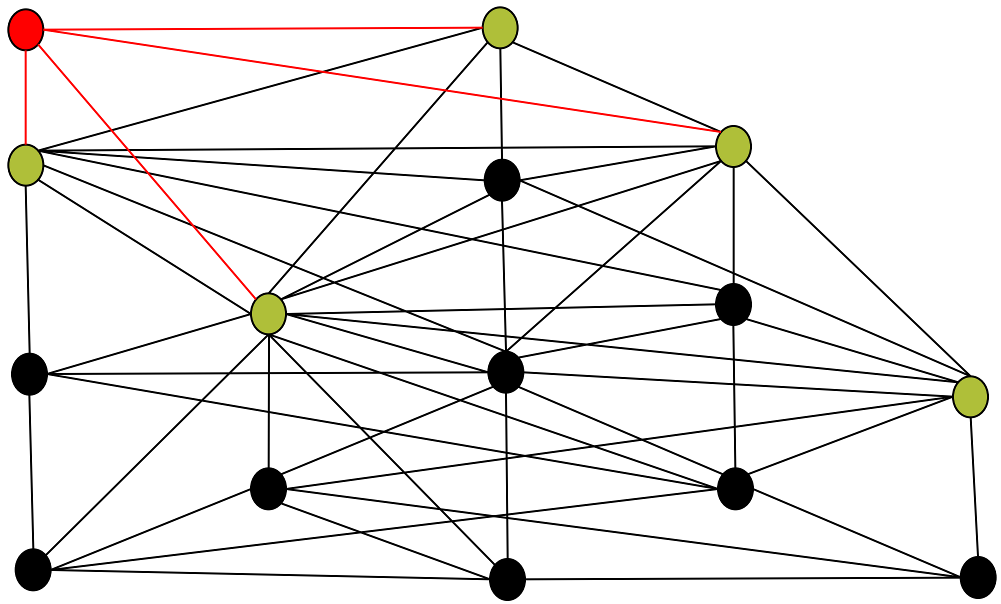
k-stack



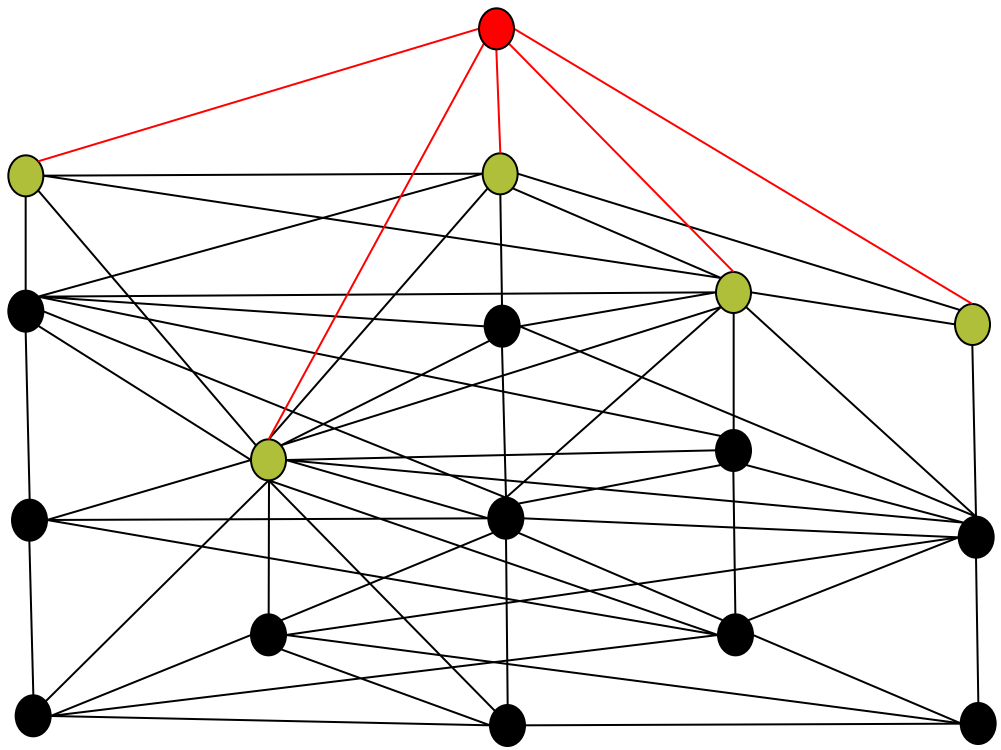
k-stack



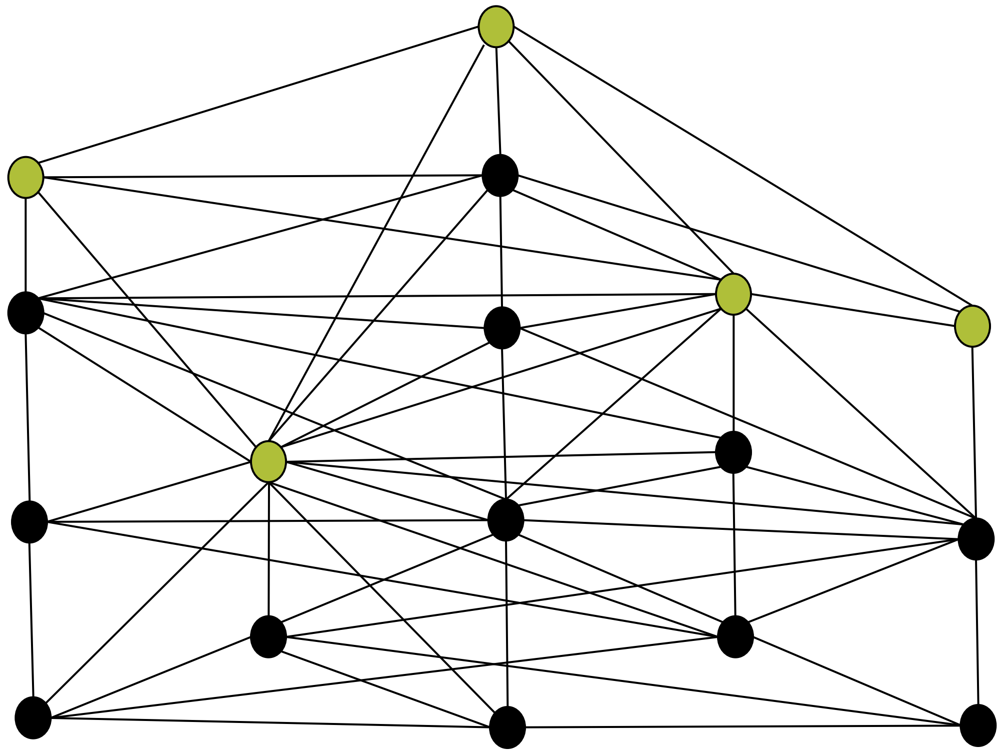
k-stack



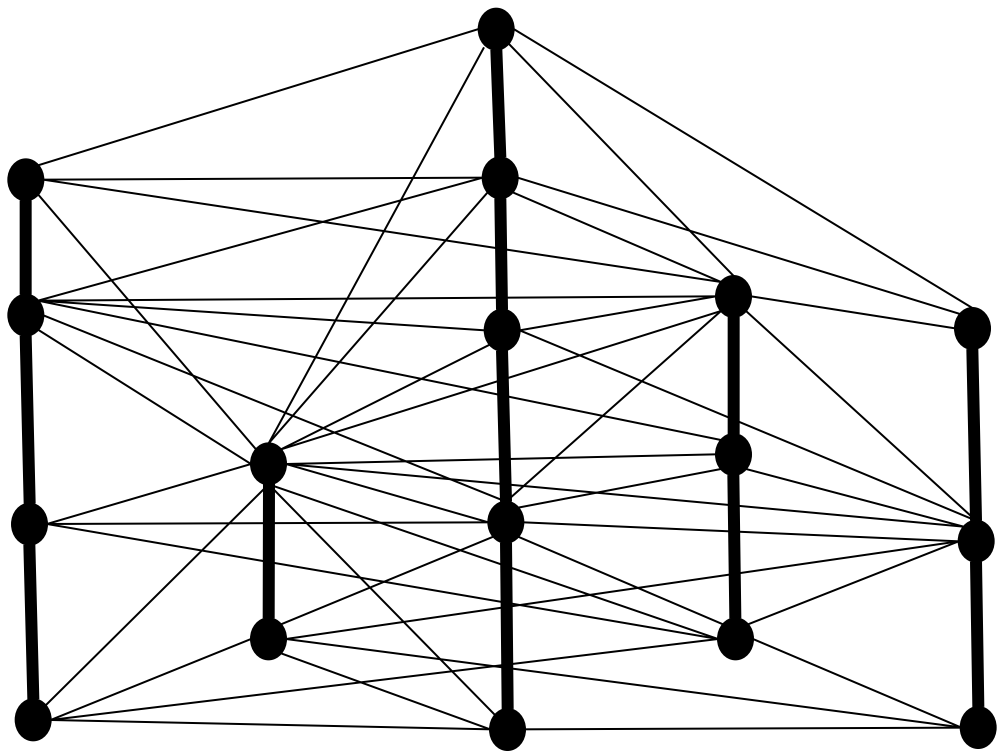
k-stack



k-stack



k-stack



Characterizations (III)

Theorem. For any connected graph G, $fms(G) = k$ if and only if G is a k-stack.

Relations to the induced-path cover

Lemma. For a graph G=(V,E) that can be cleared by k searchers in an fms-strategy S, let $V_1, \ldots, V_k$ be k subsets of V such that each vertex in V_i, $1 \le i \le k$, is visited by the same searcher in the fms-strategy S. Then $V_1, \ldots, V_k$ form a partition of V and each V_i induces a path.

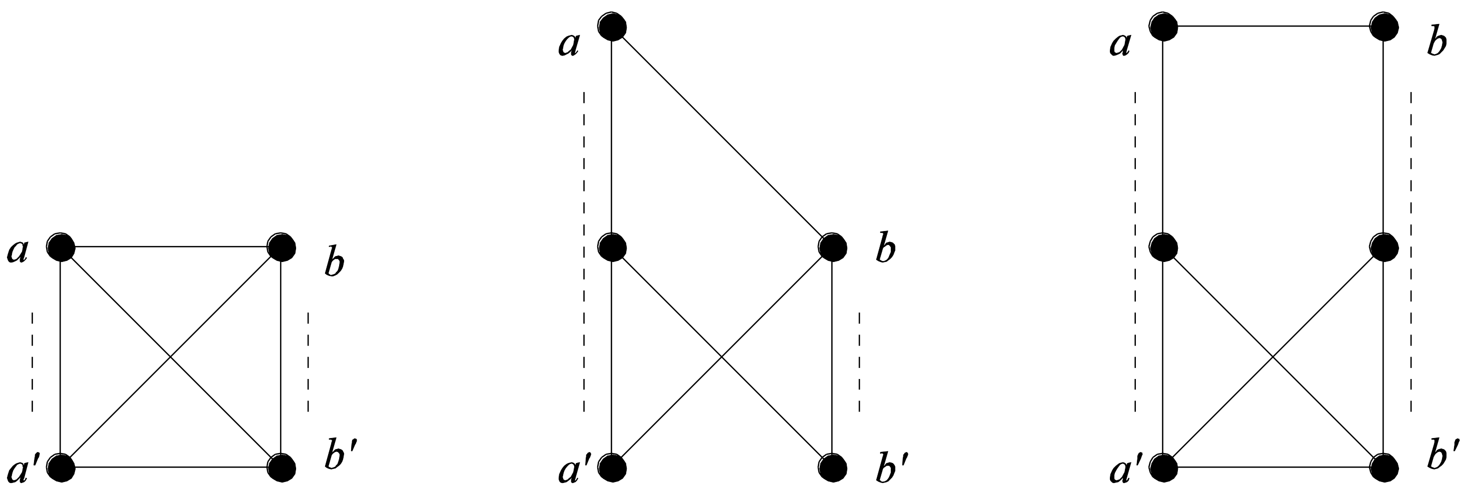
Definition. Each induced path $G[V_i]$ is called an fms-path with respect to S, and the set $G[V_1], \ldots, G[V_k]$ of fms-paths is called an fms-path cover of G with respect to S.

Relations to the induced-path cover

Theorem. For an fms-strategy S of a graph G that uses k searchers, k ≤ 2, let P be an fms-path cover of G with respect to S. For any two paths P_1 and P_2 in P, let H be the subgraph of G induced by vertices $V(P_1) \cup V(P_2)$. Then the following are equivalent:

- H does not contain any graph in the following figure, where $a, a' \in V(P_1)$ and $b, b' \in V(P_2)$.
- H has one of the three patterns:
  - (a) a forest consisting of two disjoint paths,
  - (b) a tree consisting of two adjacent degree-3 vertices and all other vertices having degree one or two; and
  - (c) a ladder.

Graphs with fms ≥ 2



Complete graphs, complete bipartite graphs and grids

- **Lemma.** For a complete graph K_n ($n \geq 2$), $fms(K_n) = n-1$.
- **Lemma.** For a complete bipartite graph $K_{m,n}$ ($n \geq m \geq 2$), $fms(K_{m,n}) = m+n-2$.
- **Lemma.** For a grid G_{mxn} with m rows and n columns ($2 \leq m \leq n$), $fms(G_{mxn}) = m$.

Trees

- Theorem. For a tree T, fms(T)=ipc(T).

- Corollary. For any tree, the fast-mixed search number and an optimal fms-strategy can be computed in linear time.

Cactus

- Theorem. For any cactus, the fast-mixed search number and an optimal fms-strategy can be computed in linear time.

Interval graphs

- Theorem. Given an interval graph G, let C_1, C_2, …, C_m be the sequence of the maximal cliques of G such that, for any v ∈ $V(C_i) \cap V(C_k)$, $1 \le i < k \le m$, the vertex v is also contained in all C_j, $i \le j \le k$. If k >1, then

$$fms(G) = |V(C_1)| + \sum_{j=1}^{m-1} \max\{|V(C_{j+1})| - |V(C_j)|, 0\}$$

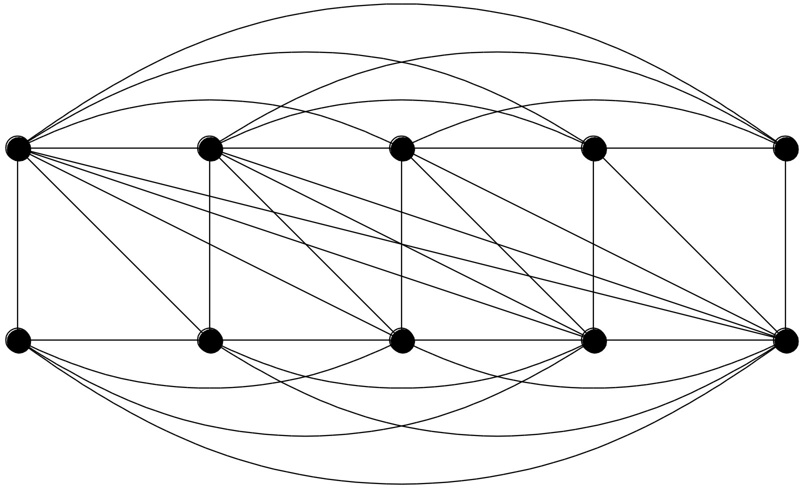
Interval graphs

- Corollary. For any interval graph, the fast-mixed search number and an optimal fms-strategy can be computed in linear time.

k-trees

- Theorem. For a k-tree G with more than k vertices, if G has exactly two simplicial vertices, then fms(G)=k.

fms-maximal graphs



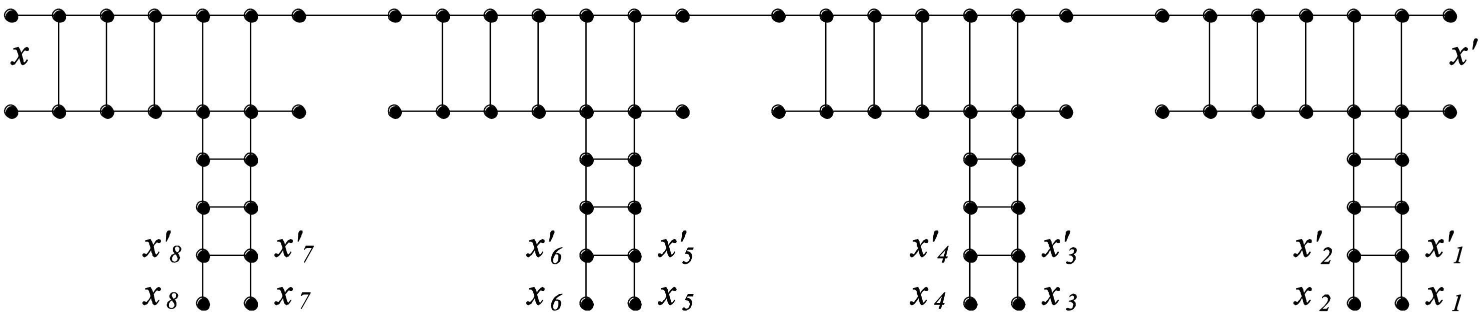
Theorem. Every fms-maximal graph G with fms(G)=k is a k-tree with exactly two simplicial vertices.

Cartesian product

- **Theorem.** For any graphs G and H,

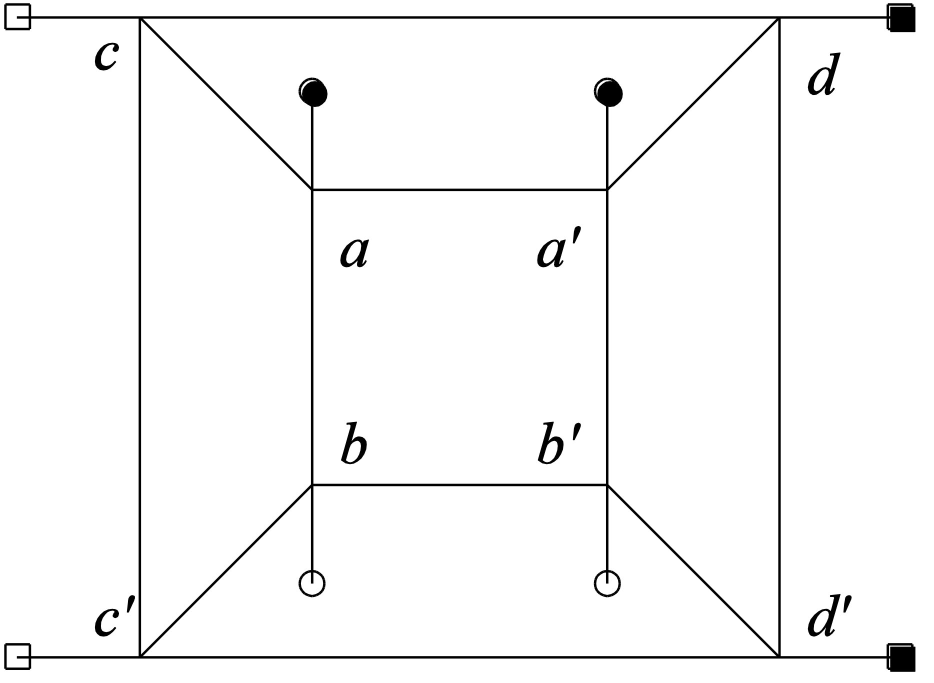
$$fms(G \square H) \leq \min\{|V(G)| fms(H), |V(H)| fms(G)\}.$$

NP-completeness



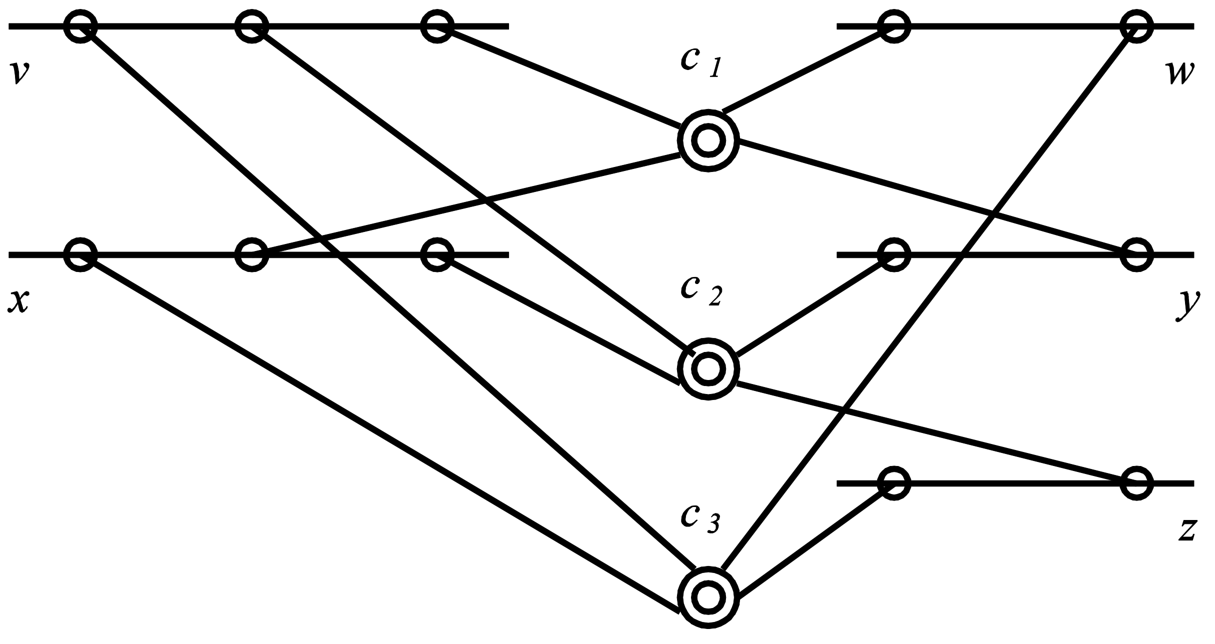
A variable gadget G^k with four legs (k=4)

NP-completeness



A clause gadget

NP-completeness



The reduction

NP-completeness

- Theorem. The fast-mixed search problem is NP-complete. It remains NP-complete for graphs with maximum degree 4.
- Corollary. Given a graph G with k leaves and maximum degree 4, the problem of determining whether fms(G)= k/2 is NP-complete.

Thank you!