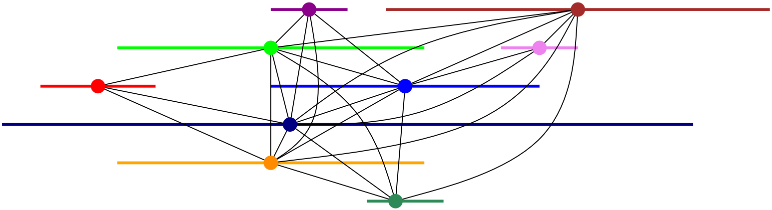
Algorithms for Scattering Number and Hamilton-Connectivity of Interval Graphs

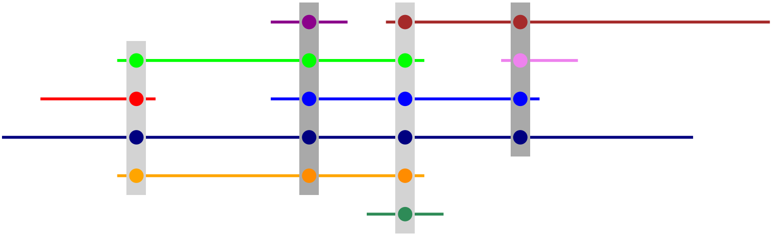
Hajo Broersma [1], Jiří Fiala [2], Petr A. Golovach [3], Tomáš Kaiser [4], Daniël Paulusma [5], Andrzej Proskurowski [6]

[1] University of Twente, The Netherlands
[2] Charles University, Czech Republic
[3] University of Bergen, Norway
[4] University of West Bohemia, Czech Republic
[5] Durham University, United Kingdom
[6] University of Oregon, U.S.A.

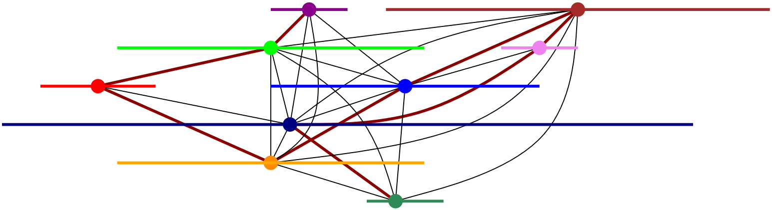
Interval graphs and their representations



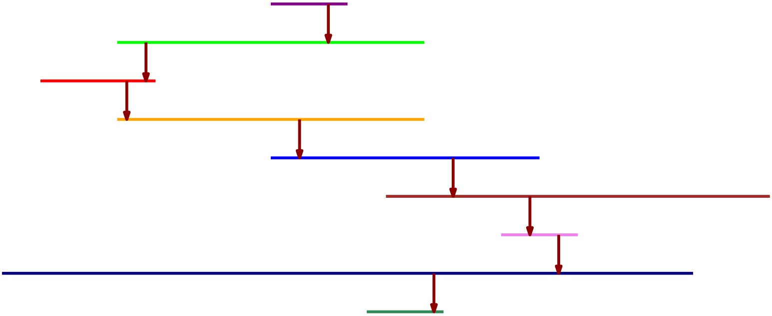
Interval graphs and their representations



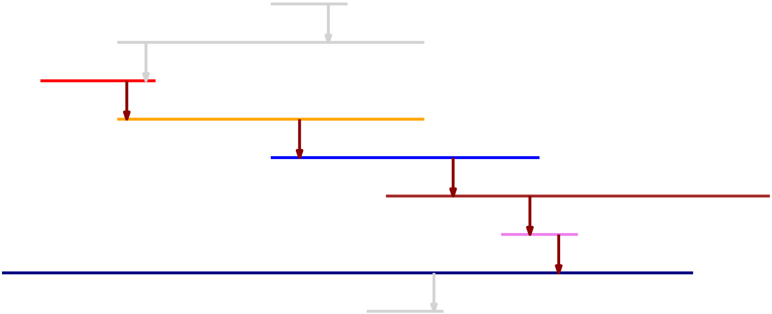
Hamiltonian paths



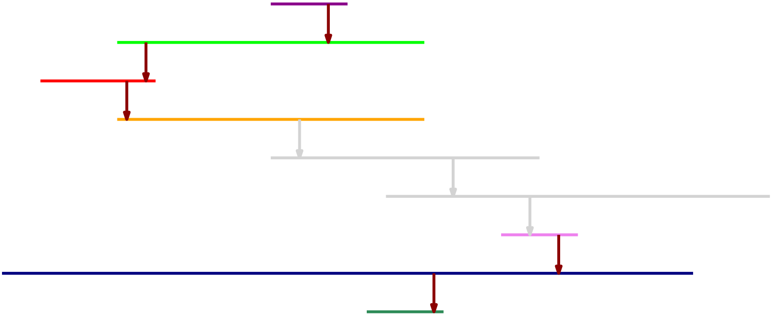
Monotonne Hamiltonian paths



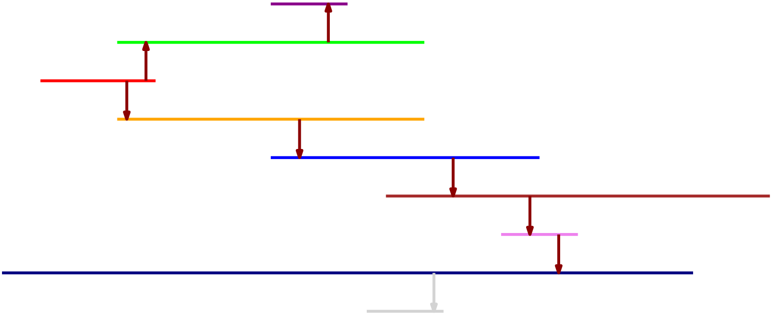
Monotonne Hamiltonian paths



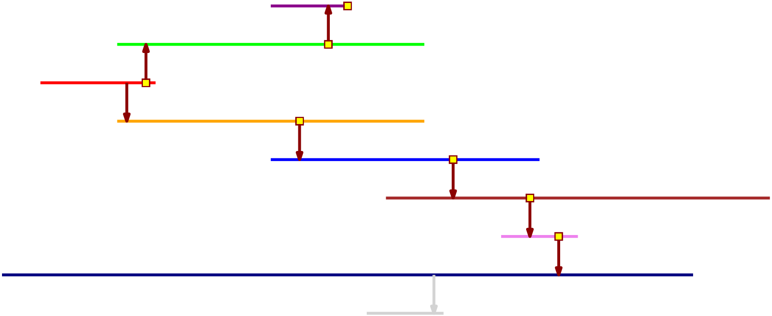
Monotonne Hamiltonian paths



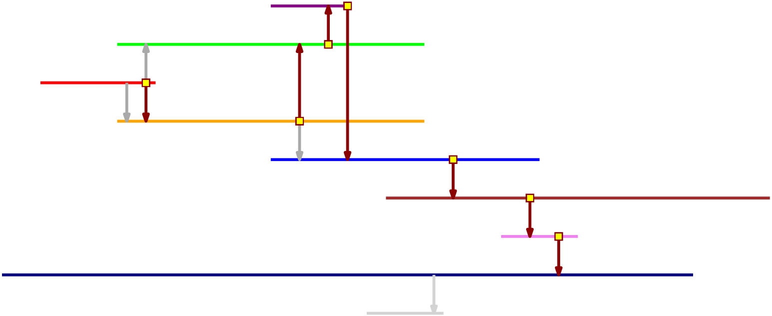
Monotonne Hamiltonian paths



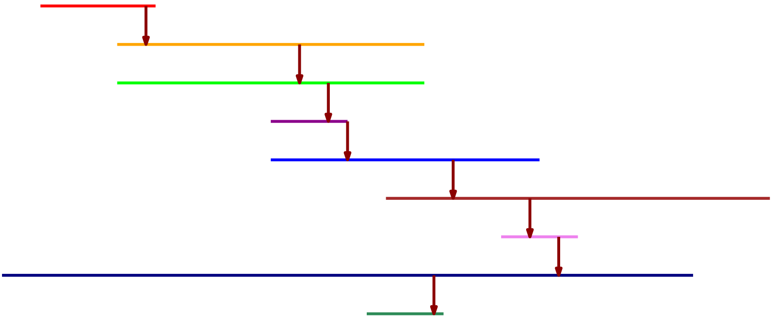
Monotonne Hamiltonian paths



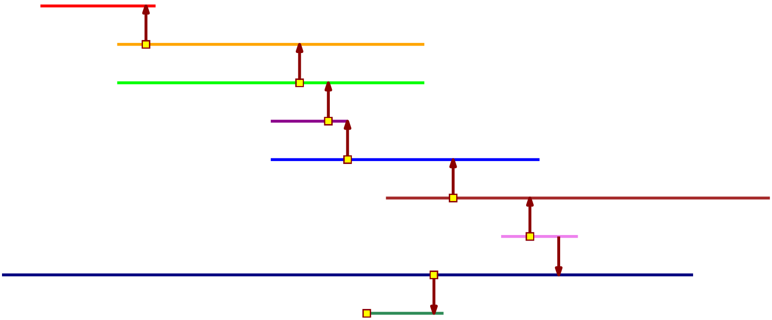
Monotonne Hamiltonian paths



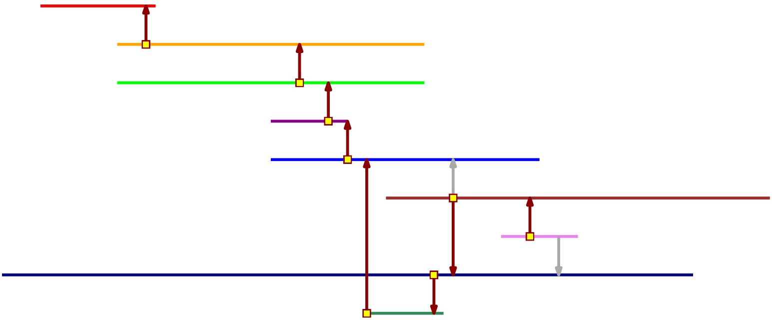
Monotonne Hamiltonian paths



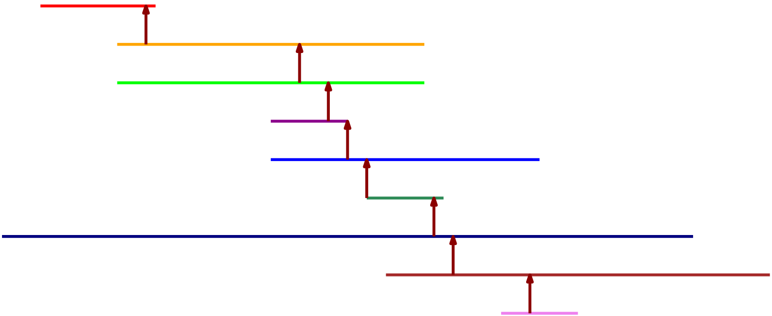
Monotonne Hamiltonian paths



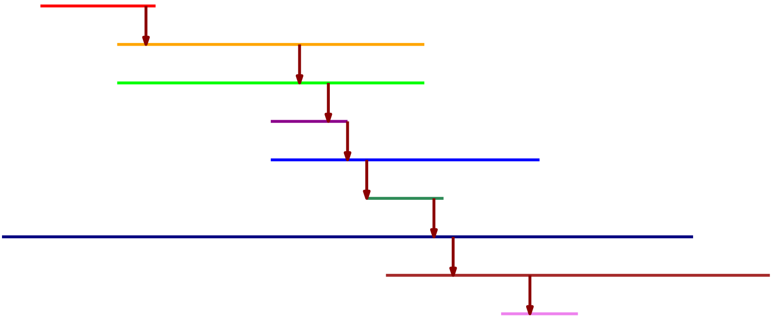
Monotonne Hamiltonian paths



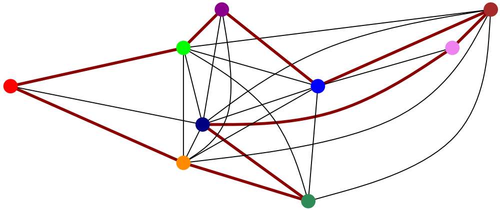
Monotonne Hamiltonian paths



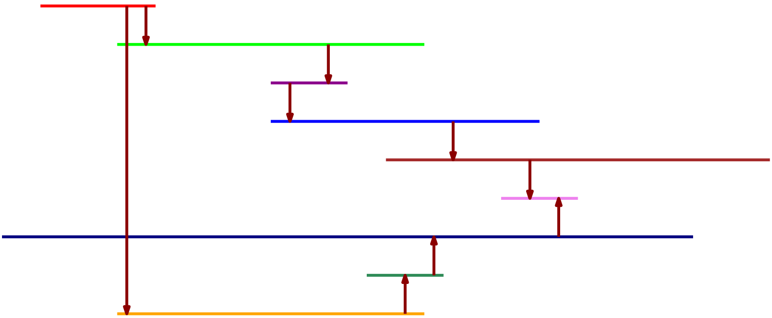
Monotonne Hamiltonian paths



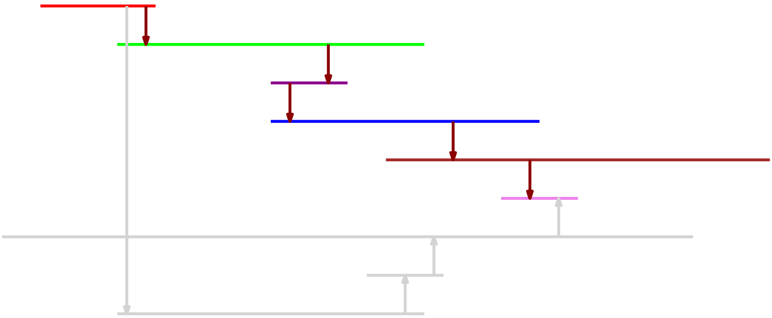
Monotonne Hamiltonian cycles



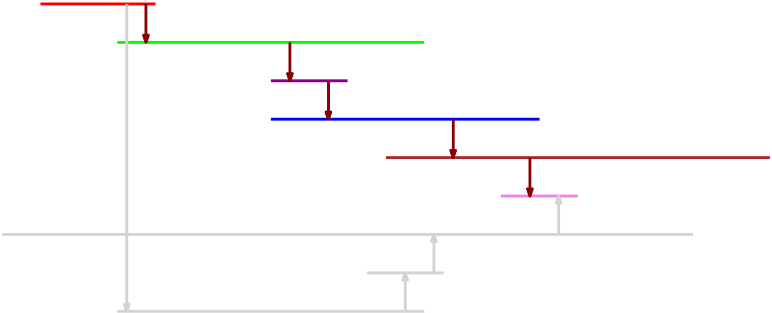
Monotonne Hamiltonian cycles



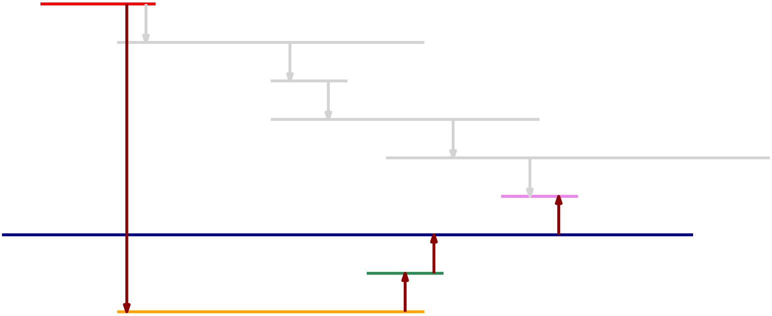
Monotonne Hamiltonian cycles



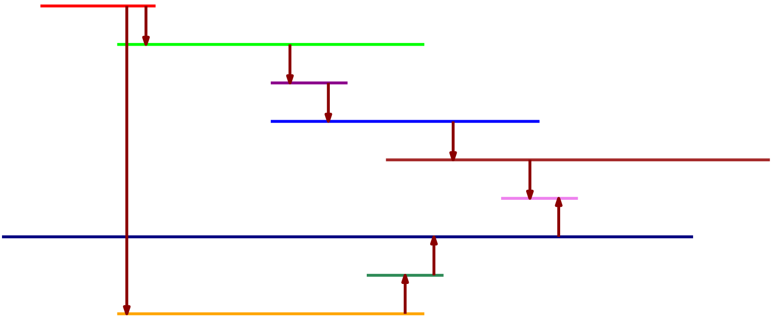
Monotonne Hamiltonian cycles



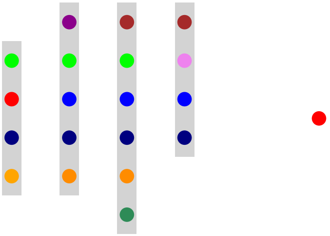
Monotonne Hamiltonian cycles



Monotonne Hamiltonian cycles

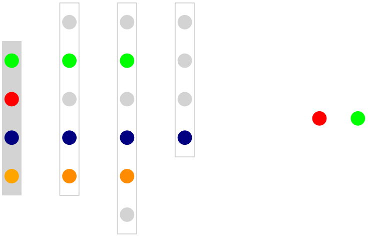


Keil's and Damaschke's greedy algorithms



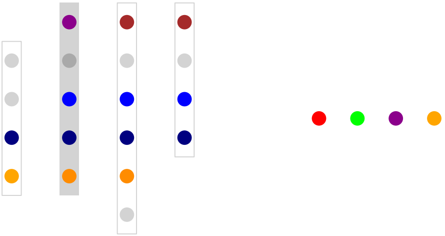
Process the cliques left to right and
extend the path by the shortest interval.

Keil's and Damaschke's greedy algorithms



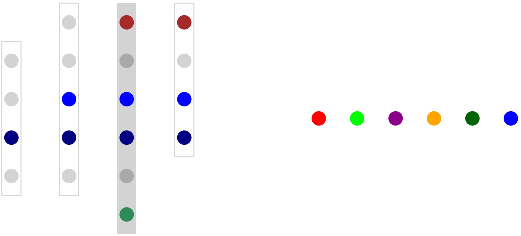
Process the cliques left to right and
extend the path by the shortest interval.

Keil's and Damaschke's greedy algorithms



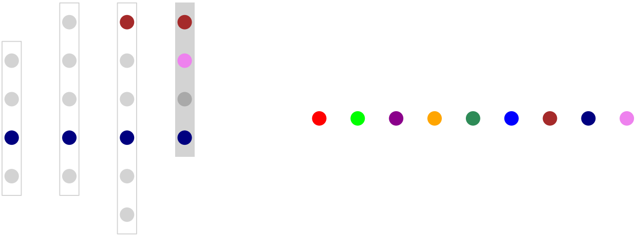
Process the cliques left to right and
extend the path by the shortest interval.

Keil's and Damaschke's greedy algorithms



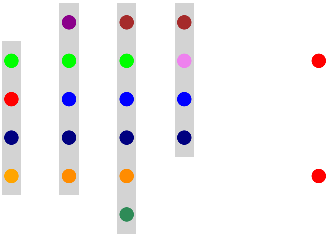
Process the cliques left to right and
extend the path by the shortest interval.

Keil's and Damaschke's greedy algorithms



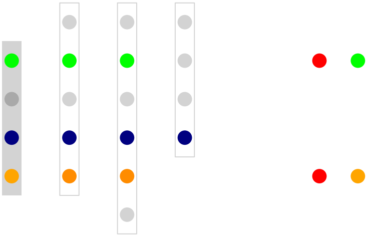
Process the cliques left to right and
extend the path by the shortest interval.

Keil's and Damaschke's greedy algorithms



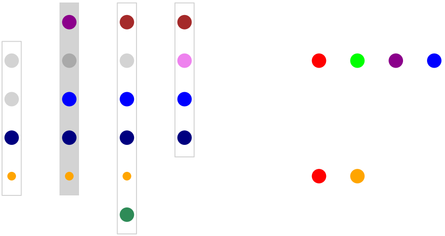
Process the cliques left to right and
extend the path by the shortest interval.

Keil's and Damaschke's greedy algorithms



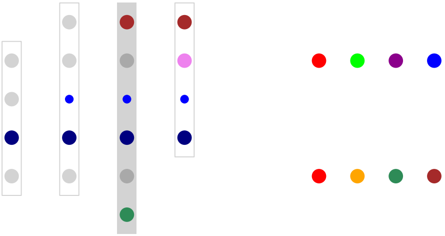
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



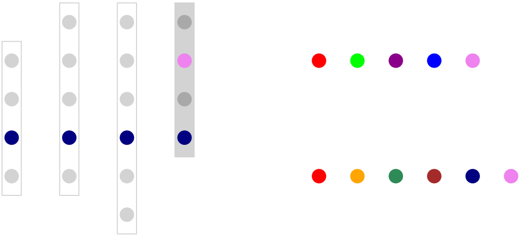
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



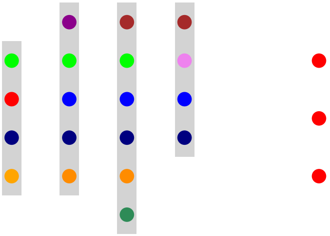
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



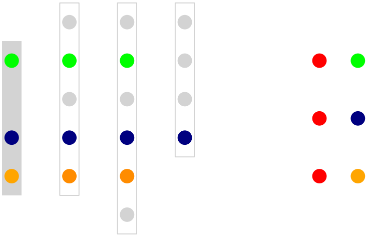
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



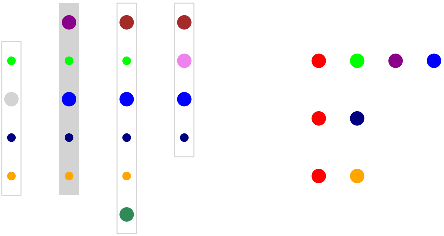
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



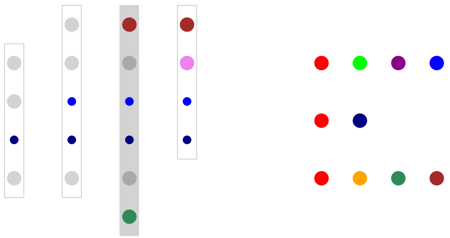
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



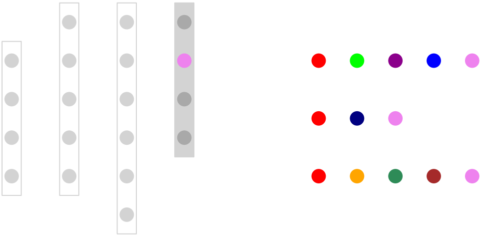
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



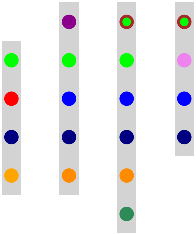
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



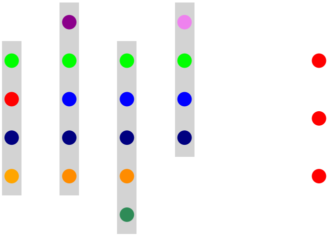
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



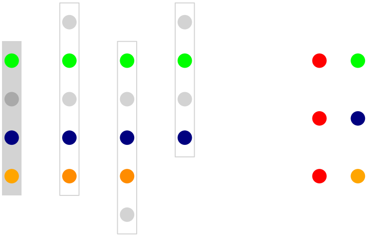
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



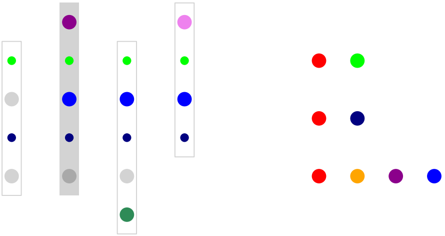
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



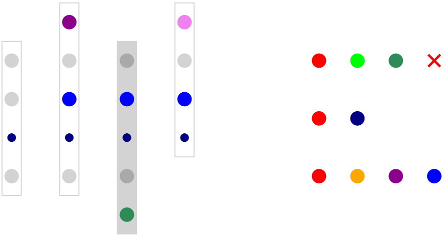
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



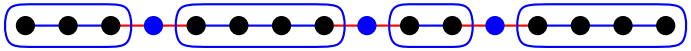
Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Keil's and Damaschke's greedy algorithms



Process the cliques left to right and
extend the path with the shortest end by the shortest interval.

Scattering number

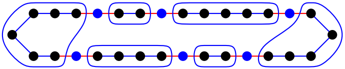


$\mathrm{scat}(G) \leq 1$

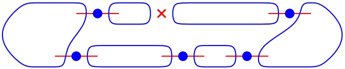
$\mathrm{scat}(G) > 2$

$$\mathrm{scat}(G) = \max_{\text{cutset } S} \mathrm{comp}(G - S) - |S|$$

Scattering number



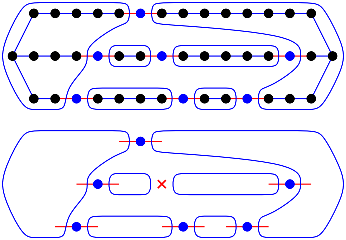
$\mathrm{scat}(G) \leq 0$



$\mathrm{scat}(G) > 1$

$$\mathrm{scat}(G) = \max_{\text{cutset } S} \mathrm{comp}(G - S) - |S|$$

Scattering number



$\mathrm{scat}(G) \leq -1$

$\mathrm{scat}(G) > 0$

$$\mathrm{scat}(G) = \max_{\text{cutset } S} \mathrm{comp}(G - S) - |S|$$

To remember

Theorem: For any interval graph G and any integer k the following are equivalent:

- $\mathrm{scat}(G) \leq k$,
- G has a Hamiltonian path for $k = 1$,
- G has a Hamiltonian cycle for $k = 0$,
- G has a spanning $(2 - k)$-stave for $k \leq 1$,
- G has a covering by at most k paths for $k \geq 1$

The idea of the algorithm for $k \leq 1$

Start with many paths.

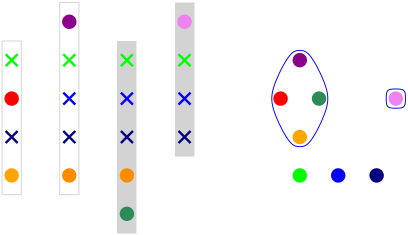
Apply the greedy algorithm.

If some path cannot be extended, continue without it.

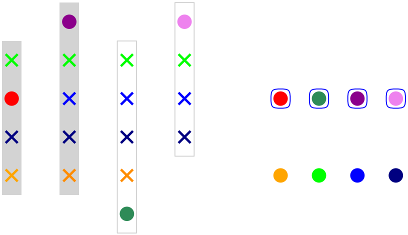
At the end merge these early finished path with any path that remain till the end.

The number of paths of the resulting spanning stave yields the scattering number: $\mathrm{scat(G)} = 2 - k$.

The construction of the scattering set by a backward analysis of the algorithm



The construction of the scattering set by a backward analysis of the algorithm

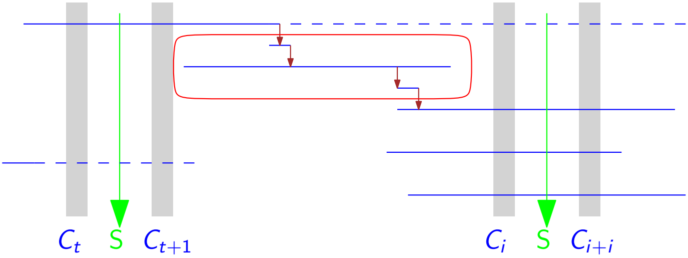


The construction of the scattering set by a backward analysis of the algorithm

If the last path was excluded during processing of the i-th maximal clique C_i, we set $S = C_i \cap C_{i+1}$ (It is the first cut of S. Now $G \setminus S$ has two components).

As long S contains a vertex, which has not been active in some cut of S, we choose the last such vertex u and determine the next cut in the form $C_t \cap C_{t+1}$, where u is u active. We include this cut into S.

When this process is finished, all vertices of S, that have not been active in the first cut, are associated with a unique component of $G \setminus S$.



k-staves and Hamiltonian connectivity

Theorem: An interval graph is k-Hamiltonian connected if and only it it contains a $(k+2)$-stave.