Measuring Indifference: Unit Interval Vertex Deletion

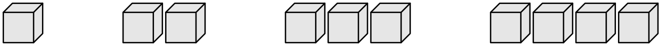
René van Bevern   Christian Komusiewicz   Hannes Moser
Rolf Niedermeier

Institut für Informatik
Friedrich-Schiller-Universität Jena, Germany

36th International Workshop on Graph-Theoretic Concepts in Computer Science, Zarós, Crete, Greece

How many sugar cubes would you like in your coffee?

The goal is placing the following alternatives in a ranking:



What is the resulting ranking if we are indifferent between i sugar cubes and $i+1$ sugar cubes?

Ranking i and $i+1$ sugar cubes equally would imply indifference between one and four sugar cubes.
⇝ Rank similar alternatives closely instead of equally.

Measuring Indifference

Approach described by Luce (1956, Econometrica), see also Aleskerov et al. (2007):

Given: Set V of objects and a set E containing $\{v, w\}$ iff a person is indifferent between $v \in V$ and $w \in V$.

Task: Find *indifference measure* $f : V \to \mathbb{R}$ with $|f(v) - f(w)| \leq \delta \iff \{v, w\} \in E$, where $\delta \in \mathbb{R}$.

Related to seriation in, e. g., archaeology (Roberts, 1971) and utility maximization in economics (Aleskerov et al., 2007).

Roberts (1969, Proof Techniques in Graph Theory): indifference measure f exists if and only if $G := (V, E)$ is a unit interval graph.

Unit Interval Vertex Deletion

If indifference measure does not exist: remove some outliers so that the remaining objects allow for indifference measure.

UNIT INTERVAL VERTEX DELETION.

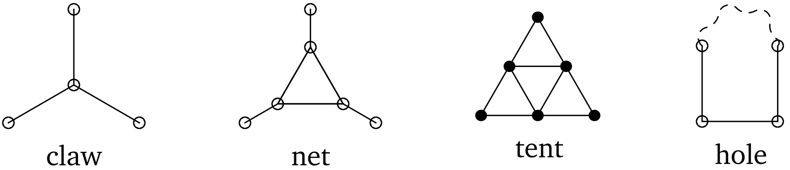
Input: A graph $G = (V, E)$ and a natural number k.

Question: Is $G - S$ a unit interval graph for some vertex set $S \subseteq V$ with $|S| \leq k$?

We call S a *unit interval vertex deletion set*.

Unit Interval Graphs

Wegner (1967, PhD thesis): unit interval graphs are precisely the graphs not containing the following induced subgraphs:



Holes are induced cycles of length greater than three.

Challenge

Marx (2010, Algorithmica) presented a fixed-parameter algorithm for the related problem:

Chordal Vertex Deletion.

Input: A graph $G = (V, E)$ and a natural number k.

Question: Is $G - S$ hole-free for some vertex set $S \subseteq V$ with $|S| \leq k$?

- extensible to solve Unit Interval Vertex Deletion
- in worst case, solves problem on tree decompositions of width $\Omega(k^4)$: running time like $2^{\Omega(k^4)} \cdot \mathrm{poly}(n)$

Main Result

Theorem. UNIT INTERVAL VERTEX DELETION *is solvable in* $O((14k+14)^{k+1} \cdot kn^6)$ *time.*

Here,

- k is the number of allowed vertex deletions and
- n is the number of vertices in the input graph.

Reduction to Simpler Problem I

To obtain structural information about the input graph, we

- destroy induced claws, nets, tents, C_4s and C_5s in the input graph using a simple search tree algorithm and
- solve Unit Interval Vertex Deletion on the resulting *almost unit interval graphs*, i. e. {claw, net, tent, C_4, C_5}-free graphs.

Proposition. Unit Interval Vertex Deletion *is NP-complete on {claw, net, tent}-free graphs.*

Reduction to Simpler Problem II

Using iterative compression due to Reed et al. (2004, Operations Research Letters), we finally arrive at

Disjoint Unit Interval Vertex Deletion.

Input: An almost unit interval graph $G = (V, E)$ and a unit interval vertex deletion set X for G.

Output: A unit interval vertex deletion set S with $|S| < |X|$ and $S \cap X = \emptyset$ or "no" if no such set exists.

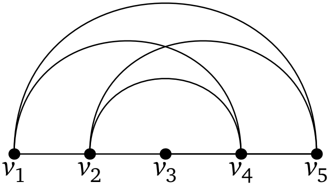
Theorem. Disjoint Unit Interval Vertex Deletion *is solvable in* $O((14|X|-1)^{|X|-1} \cdot |X|n^5)$ *time.*

Properties of Unit Interval Graphs I

One can find in linear time a *bicompatible elimination order* of the vertices of a unit interval graph (Panda and Das, 2003, IPL).

In a bicompatible elimination order, vertices of maximal cliques appear consecutively.

Example. The following order from left to right is *not* a bicompatible elimination order.

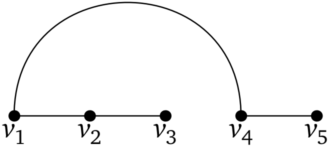


Properties of Unit Interval Graphs II

One can find in linear time a *bicompatible elimination order* of the vertices of a unit interval graph (Panda and Das, 2003, IPL).

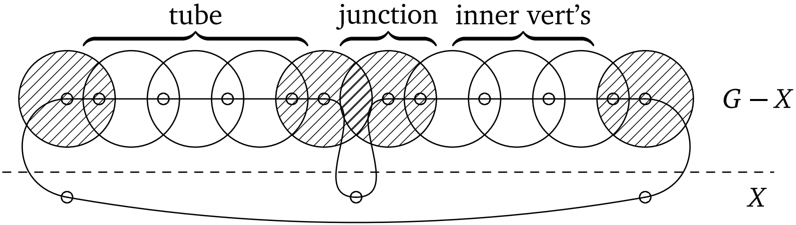
Vertices of induced paths appear in the same (or inverse) order as in a bicompatible elimination order.

Example. The following order from left to right is *not* a bicompatible elimination order.



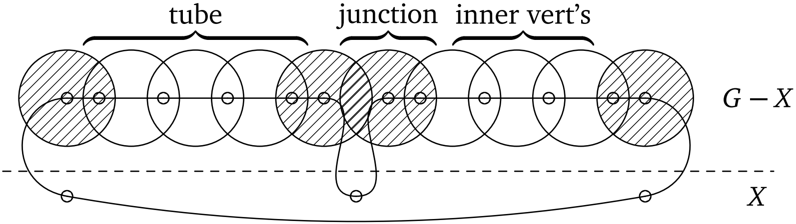
Structure of Disjoint UIVD Instance

We structure a Disjoint Unit Interval Vertex Deletion instance using a classification of the maximal cliques of $G - X$:



- Here, $G - X$ is a unit interval graph.
- If the almost unit interval graph G is not a unit interval graph, then it contains a hole of length greater than six.

Basic Idea



It remains to destroy holes of length greater than six. Idea:

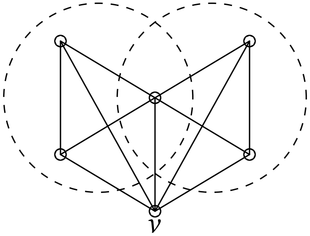
- bound the number of vertices of a hole in junctions,
- bound the number of tubes visited by a hole, finally
- show how to optimally destroy tubes in polynomial time.

Then try all possibilities of destroying a hole

- by deleting a vertex of the hole in a junction or
- by destroying a tube visited by the hole.

Properties of Almost Unit Interval Graphs

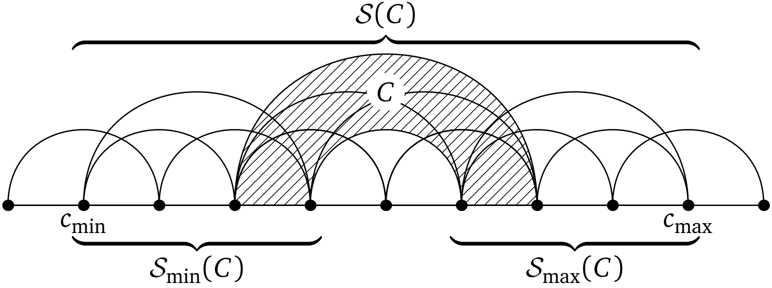
Derived from Fouquet (1993, Journal of Combinatorial Theory Series B): in an almost unit interval graph containing holes, the neighborhood of each vertex can be covered by two cliques:



Each clique contains at most two vertices of a hole.

Bounding Number of Vertices in Junctions

The neighborhood of the unit interval vertex deletion set X can be covered by $2|X|$ (not necessarily maximal) cliques. Let $\mathcal{S}(C)$ be the set of vertices of all maximal cliques intersecting one such clique C:



Observe: $\mathcal{S}(C)$ is the union of three maximal cliques and therefore contains at most six vertices of a hole.

⇝ At most $12|X|$ vertices of a hole are in junctions.

Algorithm

Input: An almost unit interval graph G and unit interval vertex deletion set X for G.

Output: A unit interval vertex deletion set smaller than X and disjoint from X.

1. Compute bicompatible elimination order for $G - X$.

2. Collect tubes in $G - X$.

3. Repeatedly find hole H in G, recursively try all possibilities of

  3.1 deleting one of $12|X|$ vertices of H in junctions and

  3.2 destroying one of $2|X| - 1$ tubes visited by H.

There are $14|X| - 1$ possibilities to destroy a hole. At most $|X| - 1$ vertices may be deleted. Running time: $O((14|X| - 1)^{|X|-1} \cdot |X| n^5)$.

Conclusion and Outlook

High polynomial running time part is due to finding nets and tents: room for improvements.

Experiments may show better performance than proven upper bound: some branching rules may delete large vertex cuts in tubes.

The algorithm is not applicable to Interval Vertex Deletion: interval graphs do not allow for bicompatible elimination orders.

The existence of a polynomial-size problem kernel is open.