Mining Chemical Graphs

Tamás Horváth

Department of Computer Science III, University of Bonn
and
Fraunhofer Institute Autonomous Intelligent Systems (IAIS)
Schloss Birlinghoven, Sankt Augustin

Germany

IAIS
Fraunhofer Institut
Intelligente Analyse- und
Informationssysteme

universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT

Learning and Mining Graph Structured Data

many real-world **machine learning/data mining** problems:

graphs: natural way of representing structural aspects of a domain

- e.g., chemical graphs, the web graph, social networks, …

- traditional machine learning/data mining algorithms:
  assume **single fixed-width table representation** of the data
  - **columns** → features
  - **rows** → objects

- graph structured objects:
  no natural single fixed-width table representation

⇨ traditional machine learning/data mining algorithms **cannot** be applied

⇨ new methods specific to graph structured objects have to be developed

Learning and Mining Graph Structured Data

most frequent **scenarios:**

i. **single-graph mining:**
objects are **(tuples of) vertices** of a single graph
- e.g., classification of webpages (vertices) in the WWW (web graph)

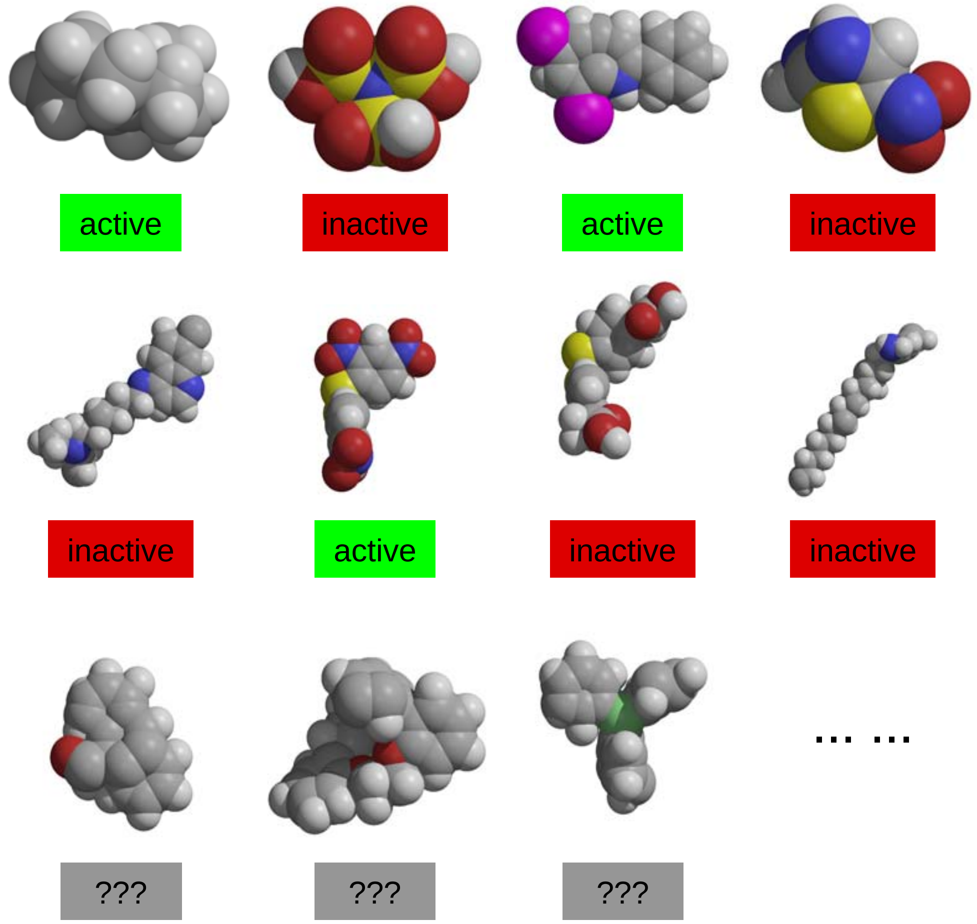
ii. **transactional graph mining:**
instances are **graphs;** elements of a graph database
- e.g., classification of chemical compounds (molecular graphs)

this talk: transactional graph mining in chemical graphs

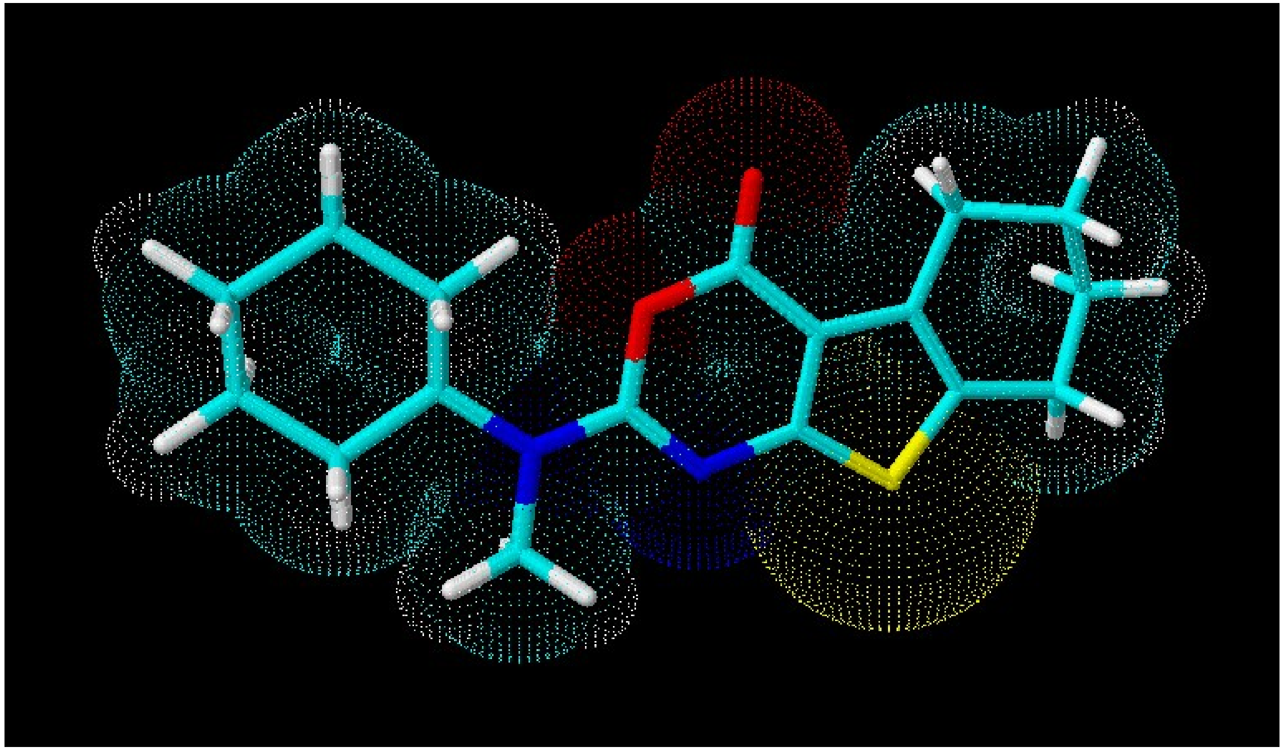
Application Example

virtual screening in drug discovery:

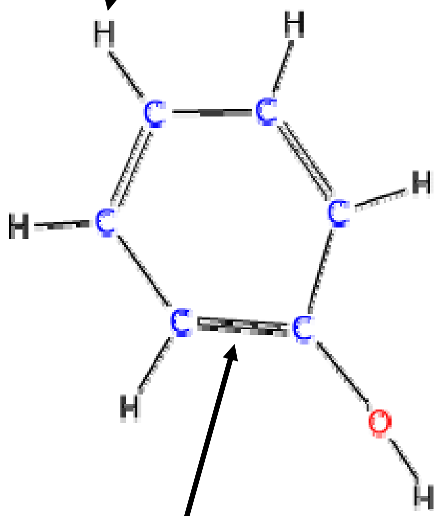
- select a limited number of candidate compounds from millions of database molecules that are most likely to possess a desired biological activity



Molecules and their Molecular Graphs



vertex label



edge label
“double”

molecules give rise to **labeled undirected graphs**

Outline

- **descriptive graph mining (local patterns)**
  1. frequent subgraph mining in *outerplanar graphs*
  2. frequent subgraph mining in graphs of *bounded treewidth*
- **conclusion**

Frequent Subgraphs

frequent graphs:

- *D*: set of labeled **graphs**,
- *t*: positive integer **threshold**,
- φ: a quasi-order (i.e., reflexive and transitive) **specialization relation** on D

- a labeled graph H is **t-frequent** w.r.t. D and φ if $|\{G \in D: H \ \varphi\ G\}| \geq t$
- *F(D,t,φ)*: **set of t-frequent graphs** w.r.t. D and φ
  - apart from *antisymmetry*, **special case** of [Mannila & Toivonen,´97]

usual cases:

- φ is the **subgraph isomorphism** (*partial order*): graph mining community
- φ is the **homomorphism**: ILP community

both cases: φ is **monotone** w.r.t. t-frequency

Enumeration Complexity

if the size of the output is **exponential** in that of the input, it is **hopeless** the algorithm to work in time **polynomial** in the size of the input

⇨ characterize the **delay time** [Johnson, Yannakakis & Papadimitriou, ´88]

I. **polynomial delay:**
   - the delay time is **always polynomial** in the **size** of the input

II. **incremental-polynomial delay:**
   - the delay time is polynomial in the **combined size** of the input and the output **so far** computed
   - after **exponentially** many steps the delay time may become **exponential**

III. **output-polynomial time:**
   - the total time is polynomial in the **combined size** of the input and the **entire** output
   - after **polynomially** many steps the delay time may become **exponential**

- **most liberal** class: **output-polynomial time**

Mining Frequent Connected Subgraphs

Given a set D of labeled graphs and an integer t ≥ 0, **enumerate** the set of t-frequent *connected* subgraphs of D w.r.t. subgraph isomorphism

- i.e. the set F(D,t,≤), where ≤ is the subgraph isomorphism

☹ cannot be solved in output-polynomial time (unless P = NP)

- can be used to decide the Hamiltonian path problem
- existing approaches resort to various heuristic strategies and restrictions of the search space (often with good empirical performance)

☺ enumerable in incremental-polynomial time if D is a set of **forests**

- [Chi, Muntz, Nijssen, & Kok, '05; survey paper]

What about problem classes beyond trees?

- **challenge for graph mining:** systematic study of graph classes and non-standard specialization operators

This Work

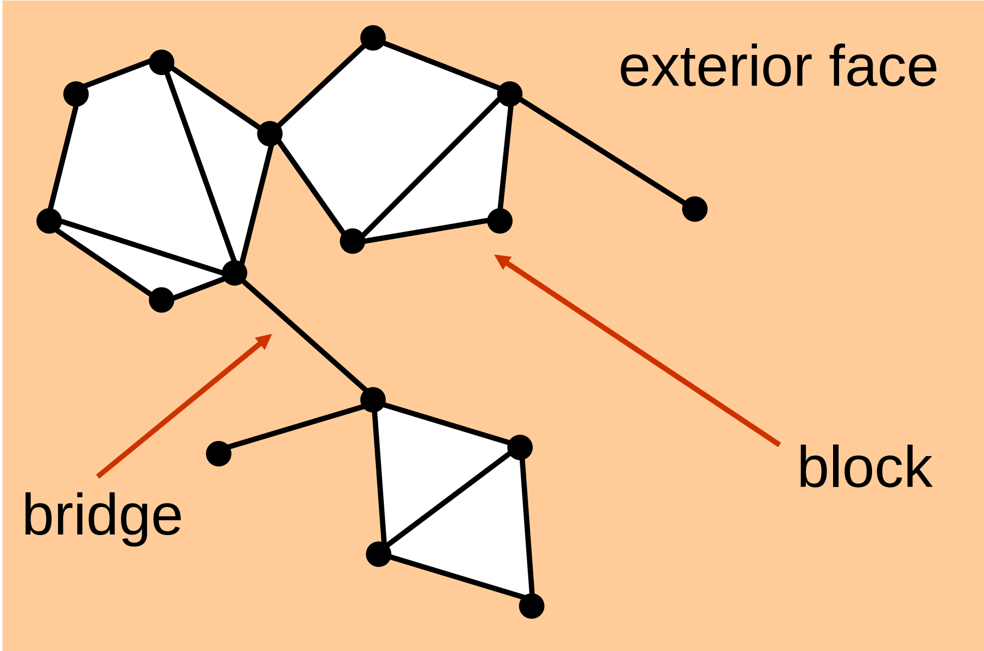
problem class:

- D: labeled **d-tenuous outerplanar** graphs (*will be defined*)
- ≤: **block and bridge preserving** subgraph isomorphism (*will be defined*)
  - **constrained** subgraph isomorphism that generalizes subtree isomorphism

Why this fragment?

1. **natural** first class beyond trees
   - trees, *outerplanar graphs*, and planar graphs form a **natural hierarchy** (Hedetniemi, Chartrand, & Geller, ´71)
2. **practically relevant** class
   - NCI dataset: **94.3%** (**236180** out of **250251**) compounds are **11-**tenuous outerplanar graphs
3. subgraph isomorphism is often **not adequate,** e.g., in chemoinformatics

Outerplanar Graphs



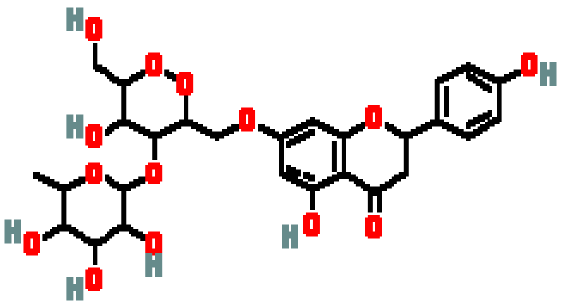
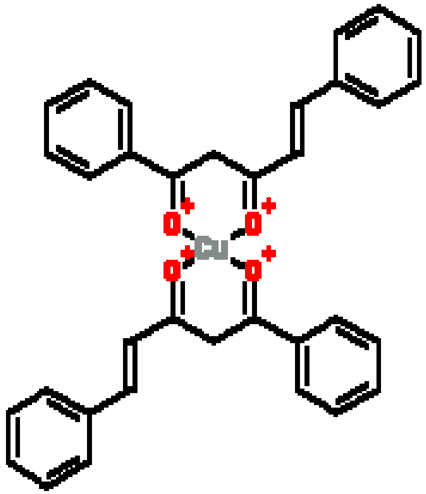
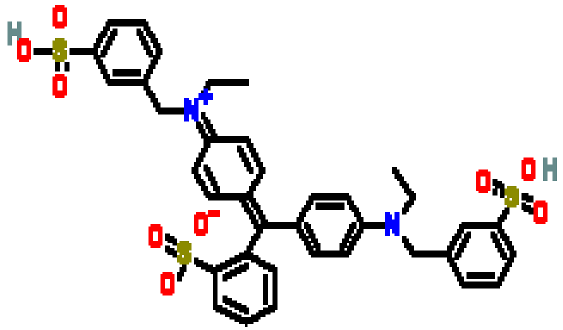
- (Chartrand & Harary, ’67)
- graphs which can be embedded in the **plane** in such a way that
  - **no two edges intersect** except at a vertex in common
  - all vertices lie on the **exterior face**

Properties:

- outerplanarity can be decided in **linear** time [Mitchell,´79]
- each block (biconnected components) with n vertices has a **unique Hamiltonian cycle**
- the unique Hamiltonian cycle
  - can be computed in **linear** time [Mitchell,´79]
  - has at most **n-3 diagonals**

d-Tenuous Outerplanar Graphs

- each block has at most **d** diagonals
- NCI dataset:
  - **236180** outerplanar graphs (out of the **250251** compounds)
  - **d = 11** (only for one compound)
  - **d = 5** for **236083** (**99.99%**) outerplanar graphs
- **d** is considered to be a **constant**!
- **some** molecular graphs from the NCI dataset:

Subgraph Isomorphism between Outerplanar Graphs

G, H outerplanar graphs;
How **hard** is to decide whether H is **subgraph isomorphic** to G?

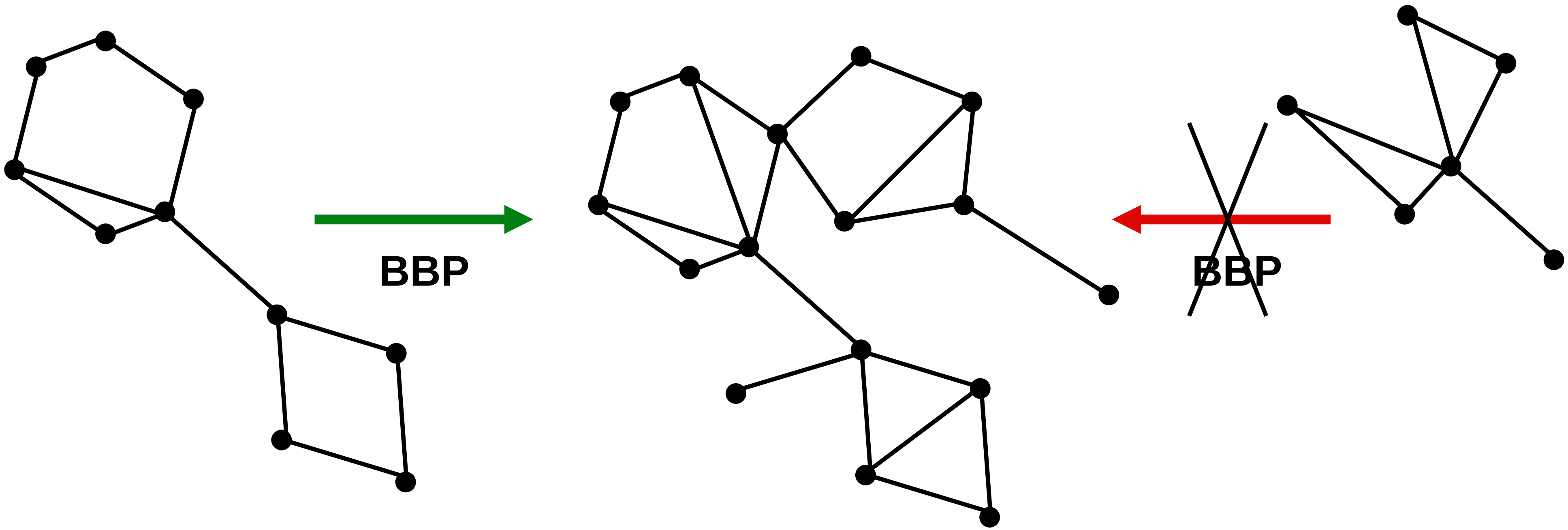
- ☹ **NP-complete** if H is **not connected**
  - generalizes the NP-complete **subforest isomorphism** problem [Garey & Johnson,´79]
- ☹ **NP-complete** even if H is connected but **not biconnected** and G is biconnected [Syslo,'82]
- ☺ decidable in time $O(|V(H)|\cdot|V(G)|^2)$ if H is **biconnected** [Lingas,´89]
  - **unlabeled** case
- ☺ decidable in time $O(|V(H)|)^{1.5}\cdot|V(G)|)$ if H and G are **trees** [Matula,´78]
  - **improved** bound $O(|V(H)|^{1.5}/\log(|V(H)|)\cdot|V(G)|)$ [Shamir &Tsur,´99]

BBP Subgraph Isomorphism

G, H outerplanar graphs;

a **block and bridge preserving (BBP)** subgraph isomorphism from H to G is a subgraph isomorphism from H to G mapping

- **different** blocks of H to **different** blocks of G
- **bridges** of H to **bridges** of G



Mining d-Tenuous Outerplanar Graphs w.r.t. BBP Subgraph Isomorphism

k-pattern: outerplanar graph s.t. number of **blocks** + number of **vertices** not belonging to any block is **k**

input: set D of d-tenuous outerplanar graphs and t > 0

1. compute the set of **frequent 1-patterns** (i.e., *vertices* + *blocks*)
2. compute the set of **frequent 2-patterns** (i.e., *edges* + *two blocks* with a common vertex + a *block* and an *edge* with a common vertex)
3. k = 2
4. **while** $L_k \neq \emptyset$ **do**
5. ++k
6. generate the set C_k of candidates from L_{k-1}
7. compute the set L_k from of frequent patterns from C_k
8. **endwhile**
9. **return** $\cup_{k>0} L_k$

Four Algorithmic Problems

1. **canonical string representation** for outerplanar graphs
   - string encoding of outerplanar graphs **unique modulo isomorphism**
   - defines a **total order** on outerplanar graphs allowing advanced data structures that support **fast search**
2. computing **frequent biconnected** outerplanar graphs
3. **candidate** generation
4. **frequency** counting

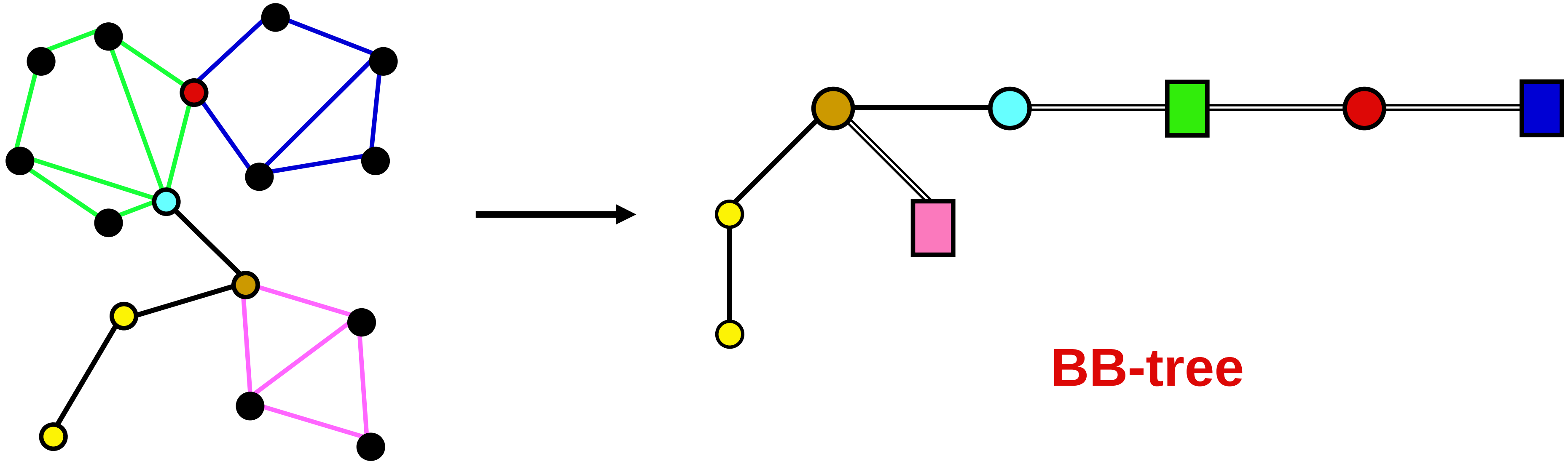
1. Canonical String Representation – BB-Trees

block and bridge graph of an outerplanar graph G

- **vertices**: bridge vertices
  \+ vertices belonging to more than one block
  \+ a vertex for each block
- **edges**: bridges of G + edges representing vertex containment

⇨ always a **free tree**

⇨ we generalize the **depth-first** canonical representation for free trees
[Chi, Muntz, Nijssen & Kok, ´05; survey]



BB-tree

2. Frequent Biconnected Outerplanar Graphs

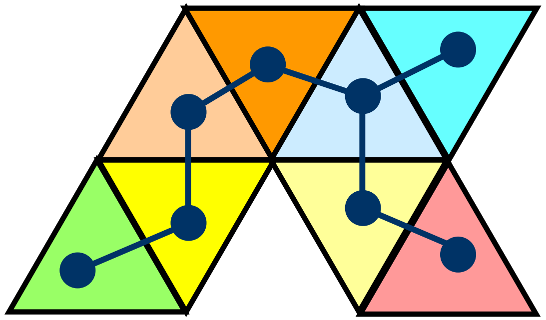
input: set D of d-tenuous outerplanar graphs, integer $t > 0$

1. compute in L_0 the set of **frequent cycles** of D
2. **for** $k = 1$ **to** d **do**
3. **let** C_k be the set of candidate biconnected graphs containing k diagonals
   - removing any diagonal results in an element of L_k
4. **let** L_k be the **frequent patterns** in C_k
5. **endfor**
6. **return** $\cup_{k=0,1,\dots,d} L_k$

2. Frequent Biconnected Outerplanar Graphs – Step 1

How to compute the **frequent cycles** in Step 1?

– a d-tenuous biconnected outerplanar graph has **at most 2^{d+1} cycles**



- always a tree
- bijection between subtrees and cycles

⇨ the number of cycles of a d-tenuous graph G is bounded by *O(|V(G)|)*

– the cycles of a graph can be enumerated with linear delay [Read & Tarjan,´75]

Lemma: For d-tenuous outerplanar graphs, the set of **frequent cycles** can be computed in time **polynomial** in the size of D.

universitätbonn
RHEINISCHE FRIEDRICH-WILHELMS-UNIVERSITÄT
Fraunhofer IAIS
Institut Intelligente Analyse- und Informationssysteme

Main Result

1. canonical string representation for outerplanar graphs ✓
2. computing frequent biconnected graphs ✓
3. candidate generation ✓
4. frequency counting ✓

Thm: Frequent d-tenuous outerplanar graphs can be enumerated in **incremental polynomial** time.

Empirical Evaluation

NCI dataset [http://cactus.nci.nih.gov/]

- most frequently used benchmark graph dataset
  - usually small subsets are considered (e.g., HIV)
- **250251** chemical graphs
  - about $\mathbf{10^7}$ compounds have so far been synthesized
- **236180** (i.e., **94.3%**) outerplanar
- max number of diagonals (**d**) is **small**:
  - **d** = **11**
  - **d = 5** for **236083** (i.e., **99.99%**)

Empirical Evaluation - Results

frequency	number of candidates	number of frequent patterns	candidate generation time (**sec**)	frequency counting time (**hours**)
10%	**925**	**521**	**0.80**	**1,98**
5%	**2688**	**1929**	**2.42**	**4,41**
2%	**36889**	**33247**	**60.08**	**12.10**
1%	**94606**	**83159**	**266.07**	**25.54**

- for 10% and 5%: **entire** set of frequent patterns
- for 2% and 1%: only the **first 18** levels

Current implementation is NOT optimized!

Outline

- **descriptive graph mining (local patterns)**
  1. frequent subgraph mining in *outerplanar graphs*
  2. frequent subgraph mining in graphs of *bounded treewidth*
- **conclusion**

Treewidth (Robertson & Seymour,'86)

- **measure** of tree-likeness of graphs

tree decomposition of a graph G:
tree T with vertices labeled by subsets of the vertex set of G s.t.

i. for each edge e of G there is a vertex of T whose label contains the vertices of e

ii. for each vertex v of G, the induced subgraph of T defined by the vertices whose labels contains v is connected (i.e., it is a tree)

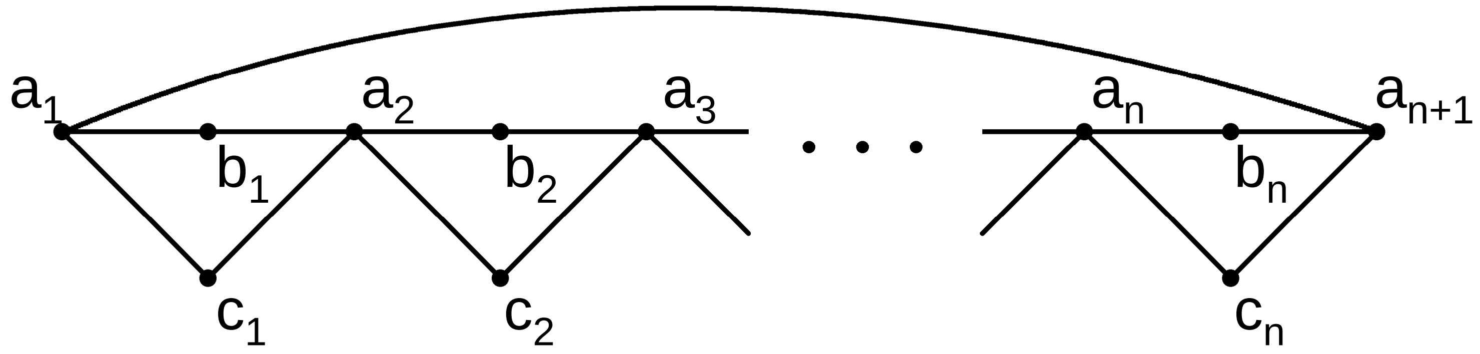
width of T:
maximum cardinality of the labels -1

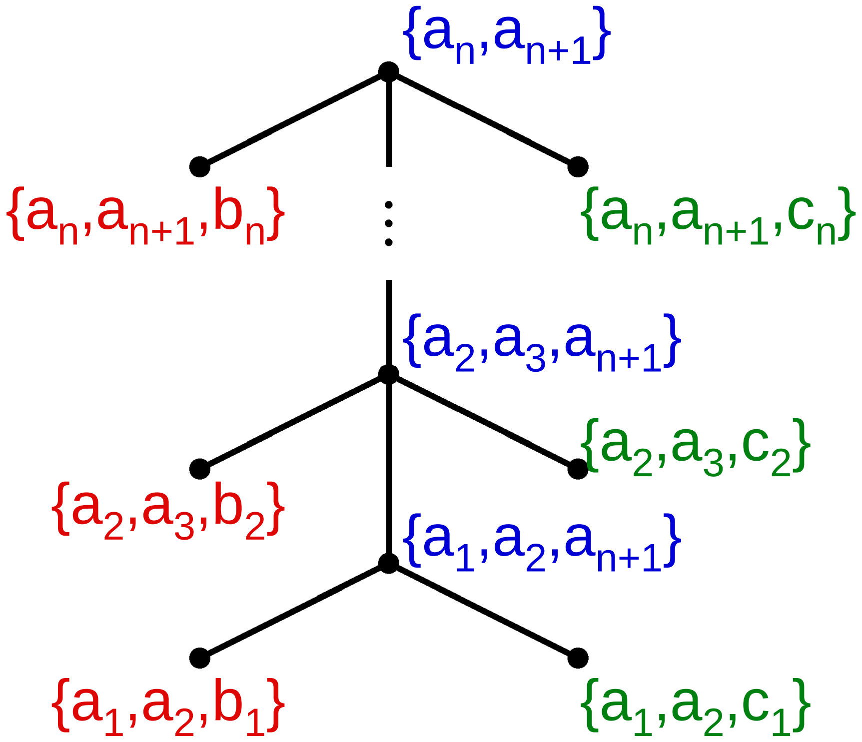
treewidth of G:
the width of a tree decomposition of G with the smallest width

- e.g., the treewidth of trees is 1; the treewidth of cycles is 2

Treewidth: Example

G:





tree decomposition of G

- treewidth: 2

treewidth of G: 2

Treewidth (cont'd)

- **useful** parameter in the design of algorithms
  - many hard problems become polynomial for graphs of bounded treewidth
- many graph classes have bounded treewidth
  - e.g., *k*-outerplanar graphs: 3*k*-1
    - ⇨ outerplanar graphs: 2
- vast majority of molecular graphs of pharmacological compounds have small treewidth
  - **e.g., NCI chemical dataset: 250251** compounds

treewidth at most 2:	**243638**	(97,36%)
treewidth at most 3:	**250186**	(99,97%)
treewidth at least 4:	**65**	(0,03%)

Mining Frequent Connected Subgraphs in Graphs of Bounded Treewidth

Given a set D of labeled graphs of treewidth at most *k* and an integer threshold t ≥ 0, **list** all *connected* graphs that are subgraph isomorphic to at least t graphs in D

- *k* is a constant

- for constant *k*, it can be decided in linear time, whether a graph has treewidth at most *k* [Bodlaender, '96]
  - not a practical result (huge hidden constant)
  - NP-complete if *k* is a parameter [Arnborg, Corneil, Proskurowski, '87]
- subgraph isomorphism remains NP-hard between graphs of treewidth at most *k*
  - NP-complete if the pattern is not k-connected **or** has more than k vertices of unbounded degree; otherwise it is tractable [Gupta & Nishimura, '96]

☹ candidate generation and test is not directly applicable

Mining Frequent Connected Subgraphs in Graphs of Bounded Treewidth

Thm [Matousek & Thomas, ’92; also Hajiaghayi & Nishimura, ‘07] for constant *k*, subgraph isomorphism between graphs of treewidth at most *k* can be decided in **polynomial time** if the pattern is *connected* and has *bounded degree*

proof: *dynamic programming algorithm*

- for the text graph, it computes a tree decomposition T of treewidth *k*
- for each node v in T, it computes a set of “properties” from v and from the properties of v’s children
  - **polynomially many**, **polynomial time computable** properties for each node in T
  - if the treewidth is only restricted then the number of properties can be **exponential**

Mining Frequent Connected Subgraphs in Graphs of Bounded Treewidth

Thm: can be solved in incremental polynomial time

proof idea:

- levelwise (BF-search) generation of candidate patterns
  - add one new edge to a pattern s.t. the graph obtained has treewidth at most k
- to decide whether a candidate pattern P is subgraph isomorphic to a transaction graph in D, it is sufficient to compute a set of properties with cardinality polynomial in the **combined size** of D and the set of frequent patterns listed before P
  - the delay can be exponential only after the enumeration of exponentially many frequent patterns

Conclusion and Future Work

frequent connected subgraph mining in outerplanar graphs:

- **positive** result for a **practically relevant** graph class **beyond trees**
- BBP subgraph isomorphism algorithm **may be of interest in itself**

frequent connected subgraph mining in graphs of bounded treewidth:

- positive result though the **matching operator is NP-complete**
  - ⇨ *efficient pattern mining is possible even for NP-hard matching operators!*

? enumerable with **polynomial delay**

- design and implementation of a practical algorithm for graphs of treewidth at most 3
  - vast majority of molecular graphs of pharmacological compounds have treewidth at most 3

Acknowledgements to Coauthors

Jan Ramon

K.U. University of Leuven

Stefan Wrobel

Fraunhofer IAIS

University of Bonn