**Efficient Frequent Connected Induced Subgraph Mining in Graphs of Bounded Tree-width**

Tamás Horváth¹,², Keisuke Otaki³, Jan Ramon⁴

¹ Dept. of Compute Science, University of Bonn, Germany
² Fraunhofer IAIS, Schloss Birlinghoven, Sankt Augustin, Germany
³ Dept. of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Kyoto, Japan
⁴ Dept. of Computer Science, Katholieke Universiteit Leuven, Belgium

---

**General setting for frequent pattern mining in graphs: Given**

- A database 
  \[ D = \{G_1, \ldots, G_n\} \]
  with 
  \[ G_i \in \mathcal{G} \]
  for some graph class \( \mathcal{G} \),
  a pattern language \( \mathcal{P} \),
  a matching operator \( \Rightarrow \),
  a frequency threshold \( t \in \mathbb{N} \),

**list all frequent patterns**, i.e., all patterns \( P \in \mathcal{P} \) satisfying

\[ \left| \{ G \in D : P \Rightarrow G \} \right| \geq t \]

**thm:** Computational intractable if no restriction on \( \mathcal{G} \)

**main result:** The frequent pattern mining problem can be solved in incremental polynomial time for the following problem setting:

- \( \mathcal{G} \): bounded tree-width graphs,
- \( \mathcal{P} \): connected bounded tree-width graphs,
- \( \Rightarrow \): induced subgraph isomorphism

---

**Significance of the Result**

Theoretical:
- Computationally intractable pattern matching operators do not imply the intractability of frequent pattern mining problems
  - induced subgraph isomorphism is NP-complete
    even for graphs of bounded tree-width 2

Practical:
- Graphs of small tree-width form a practically relevant graph class
  - e.g., 99.99% of the chemical graphs in the ZINC dataset (~ 16.5 million compounds) have tree-width at most 3

---

**Problem and Main Result**

- **General setting for frequent pattern mining in graphs:** Given
- **list all frequent patterns**, i.e., all patterns \( P \in \mathcal{P} \) satisfying
  \[ \left| \{ G \in D : P \Rightarrow G \} \right| \geq t \]
- **thm:** Computational intractable if no restriction on \( \mathcal{G} \)
- **main result:** The frequent pattern mining problem can be solved in incremental polynomial time for the following problem setting:
  - \( \mathcal{G} \): bounded tree-width graphs,
  - \( \mathcal{P} \): connected bounded tree-width graphs,
  - \( \Rightarrow \): induced subgraph isomorphism

---

**Notions**

- **P is induced subgraph isomorphic to G** if \( G \) has and induced subgraph isomorphic to \( P \)
- A graph \( G \) with \( n \) vertices is a **k-tree** if one of the following conditions holds:
  - \( n = k \) and \( G \) is a clique, or
  - \( n > k \) and \( G \) can be obtained from a \( k \)-tree \( G' \) with \( n-1 \) vertices by introducing a new vertex \( v \) and connecting \( v \) with all vertices of a \( k \)-clique of \( G' \)
- A graph of **tree-width at most k:** subgraph of \( k \)-tree
- Give rise to certain trees, called **tree-decompositions**
  - Many NP-hard problems on arbitrary graphs can be decided **efficiently** for graphs of bounded tree-width by dynamic programming algorithms operating on tree-decompositions
  - **Bad news:** induced subgraph isomorphism is NP-complete for \( tw = 2 \)

Note on listing complexity:
- **Polynomial delay:** Durations of printing outputs are bounded by polynomial of the size of the input
- **Incremental polynomial delay:** Durations of printing outputs are bounded by the combined size of the input and the set of outputs printed before the current pattern