Parameterized complexity of fair deletion problems II.

Dušan Knop, Tomáš Masařík, Tomáš Toufar

Faculty of Mathematics and Physics,
Charles University,
Prague, Czech Republic.

GROW 2017,
Toronto, Canada

CSI
IÚUK

Deletion problems

Given a graph property P and graph G, vertex (edge) deletion problem is a task of finding set S of vertices (edges) such that $G \setminus S$ satisfies P.

Deletion problems

Given a graph property P and graph G, vertex (edge) deletion problem is a task of finding set S of vertices (edges) such that $G \setminus S$ satisfies P.

Examples:

- Vertex Cover – $W \subseteq V$ such that $G \setminus W$ has no edges.
- Feedback Vertex set – $W \subseteq V$ such that $G \setminus W$ is a forest.
- Feedback Arc set – $F \subseteq E$ such that $G \setminus F$ is a DAG.
- Odd cycle transversal – $W \subseteq V$ such that $G \setminus W$ is a bipartite.
- Odd edge cycle transversal – $F \subseteq E$ such that $G \setminus F$ is a bipartite.

For monotone properties finding any such set is trivial.
Usual aim is to find the smallest such set.

Fair deletion problems

Usually aim is to find smallest set S such that $G \setminus S$ satisfies P.

In Fair deletion problems, we want to find set that is “locally” small.

- For a set F of edges, we want to minimize

$$\max_{v \in V} \deg_F(v).$$

- For a set W of vertices, we want to minimize

$$\max_{v \in V} |N(v) \cap W|.$$

Parameterized complexity

In parameterized complexity in addition to the input, we have a number called parameter.
Examples of parameters:

- size of the solution
- structural parameters (treewidth, clique width, vertex cover...)

Parameterized complexity

In parameterized complexity in addition to the input, we have a number called parameter.
Examples of parameters:

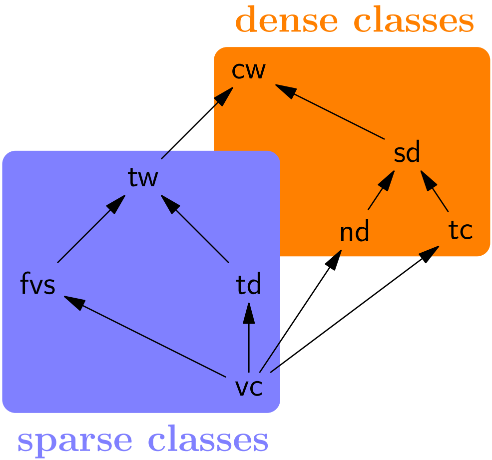
- size of the solution
- structural parameters (treewidth, clique width, vertex cover...)

The running time is described as a function of both the size of the input and the parameter.

Complexity classes

- FPT – class of problems solvable in time $f(k)n^c$
- XP – class of problems solvable in time $n^{f(k)}$
- W[1]-hard –class of problems that unlikely admit an FPT alg.

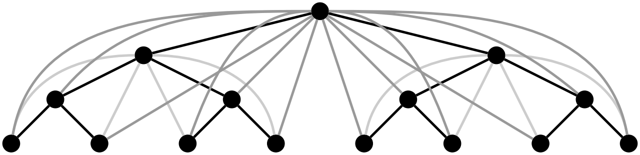
Overview of structural parameters



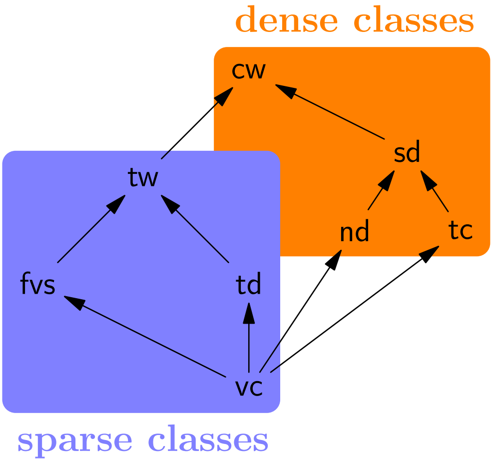
Tree depth

Definition (Tree depth)

- The closure $Clos(F)$ of a forest F is the graph obtained from F by making every vertex adjacent to all of its ancestors.
- The tree depth, denoted as $\mathrm{td}(G)$, of a graph G is one more than the minimum height of a rooted forest F such that $G \subseteq Clos(F)$.



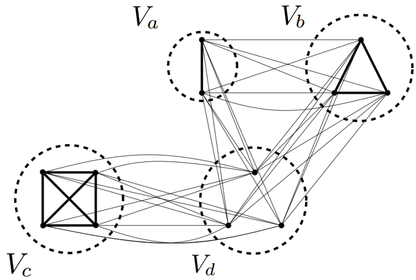
Overview of structural parameters



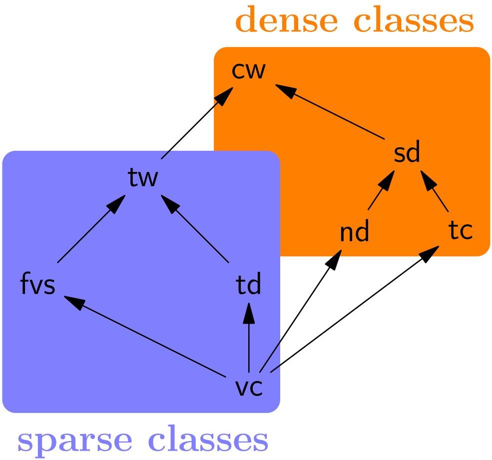
Neighborhood diversity

Definition (Neighborhood diversity)

The neighborhood diversity of a graph G is denoted by $\mathrm{nd}(G)$ and it is the minimum size of a partition of vertices into classes such that all vertices in the same class have the same neighborhood, i.e. $N(v) \setminus \{v'\} = N(v') \setminus \{v\}$, whenever v, v' are in the same class.



Overview of structural parameters



Graph properties

We study properties definable in graph logic (FO, MSO_1, MSO_2).

Graph properties

We study properties definable in graph logic (FO, MSO_1, MSO_2).

Sometimes we want to put additional restriction on the deleted set itself (for example Connected vertex cover)
We use an MSO formula with one free set variable S such that $G \models \varphi(S)$ (in contrast to original $G \setminus S \models \psi$).

We call that generalized deletion problem or some authors use monadic second order evaluation.

Fair Vertex Cover

Problem formulation

Formulated as a Fair Vertex Deletion problem: Find a set of vertices W such that the rest of the graph has no edges.

Fair Vertex Cover

Problem formulation

Formulated as a Fair Vertex Deletion problem: Find a set of vertices W such that the rest of the graph has no edges.

Our results

- FPT algorithm for the Fair vertex cover problem parameterized by modular width.
  (D. Knop, TM, T. Toufar 2017+)
- The Fair vertex cover problem is W[1]-hard parameterized by tree depth + feedback vertex set.
  (D.Knop, TM, T. Toufar 2017+)

Known results

- An XP algorithm for the generalized version of the Fair MSO_2 edge deletion problem parameterized by treewidth ($f(|\varphi|)n^{O(\mathrm{tw}(G))}$). Kolman, Lidický, and Sereni
- Can be easily adapted to the vertex version.

Known results

- An XP algorithm for the generalized version of the Fair MSO_2 edge deletion problem parameterized by treewidth $(f(|\varphi|)n^{O(\mathrm{tw}(G))})$. Kolman, Lidický, and Sereni
- Can be easily adapted to the vertex version.
- MSO_2 does not admit XP algorithm on cliques unless standard complexity assumption fails by Lampis or Courcelle, Makowsky and Rotics.

Our results — Vertex deletion problem

Positive results

- FPT algorithm for generalized version of Fair MSO_1 vertex deletion problem parameterized by neighbourhood diversity. (TM, T. Toufar 2017)
- FPT algorithm for generalized version of Fair MSO_1 vertex deletion problem parameterized by twin cover. (D. Knop, TM, T.Toufar 2017+)

Our results — Vertex deletion problem

Positive results

- FPT algorithm for generalized version of Fair MSO_1 vertex deletion problem parameterized by neighbourhood diversity. (TM, T. Toufar 2017)
- FPT algorithm for generalized version of Fair MSO_1 vertex deletion problem parameterized by twin cover. (D. Knop, TM, T.Toufar 2017+)

Hardness results

- The Fair vertex cover problem is W[1]-hard parameterized by tree depth + feedback vertex set. (D.Knop, TM, T. Toufar 2017+)

Our results — Edge deletion problem

Positive results

- FPT algorithm for generalized version of Fair MSO_2 edge deletion problem parameterized by vertex cover.
  (TM, T. Toufar 2017)

Our results — Edge deletion problem

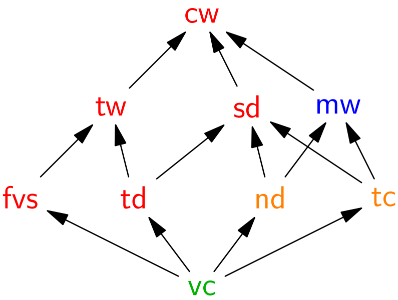
Positive results

- FPT algorithm for generalized version of Fair MSO_2 edge deletion problem parameterized by vertex cover.
  (TM, T. Toufar 2017)

Hardness results

- FO Fair deletion is W[1]-hard with respect to tree depth + feedback vertex set size.
  (TM, T. Toufar 2017)

Overview of the results



Green means FPT algorithm for MSO_2 edge deletion problem.
Orange means FPT algorithm only for MSO_1 vertex deletion problem.
Red means hardness results for both edge and vertex deletion problem.
Blue means it is mostly unknown.

Open questions

- Is there a “basic” edge deletion problem such that it is W[t]-hard on graphs of bounded tree depth, feedback vertex set or shrub depth?

Open questions

- Is there a “basic” edge deletion problem such that it is W[t]-hard on graphs of bounded tree depth, feedback vertex set or shrub depth?
- Are there other structural parameters where we can obtain FPT algorithm? (e.g. modular width)

Open questions

- Is there a “basic” edge deletion problem such that it is W[t]-hard on graphs of bounded tree depth, feedback vertex set or shrub depth?
- Are there other structural parameters where we can obtain FPT algorithm? (e.g. modular width)
- Are there NP-hard fair vertex deletion problems that admit an FPT algorithm parameterized by tree depth (and feedback vertex set)?

Open questions

- Is there a "basic" edge deletion problem such that it is W[t]-hard on graphs of bounded tree depth, feedback vertex set or shrub depth?
- Are there other structural parameters where we can obtain FPT algorithm? (e.g. modular width)
- Are there NP-hard fair vertex deletion problems that admit an FPT algorithm parameterized by tree depth (and feedback vertex set)?

Thank you for your attention!