Parameterized complexity of fair deletion problems

Dušan Knop, Tomáš Masařík, Tomáš Toufar

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

CSGT 2017,
Hejnice, Czech Republic

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 edge.
- 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

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

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)
One possibility: MSO formula with one free set variable, goal is to find set 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.

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, cliquewidth, vertex cover...)

Parameterized complexity

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

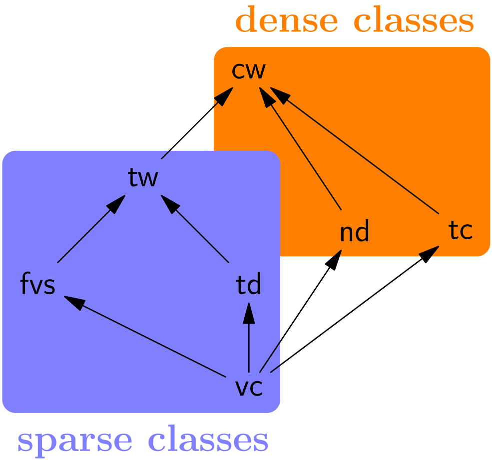
- size of the solution
- structural parameters (treewidth, cliquewidth, vertex cover…)

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



Known results

- XP algorithm for generalized version of 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

- XP algorithm for generalized version of 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 is XP-hard on cliques unless 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 treedepth. (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

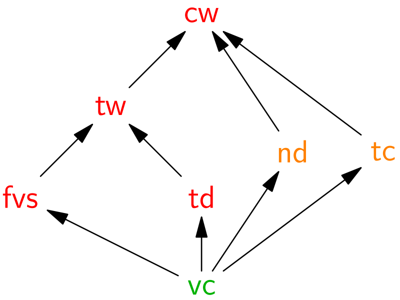
Positive results

- 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 treedepth + 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.

Open questions

- Is there a ‘basic’ edge deletion problem such that it is W[t]-hard on graphs of bounded treedepth?

Open questions

- Is there a ‘basic’ edge deletion problem such that it is W[t]-hard on graphs of bounded treedepth?
- Are there other structural parameters where can we obtain FPT algorithm? (eg. modularwidth)

Open questions

- Is there a 'basic' edge deletion problem such that it is W[t]-hard on graphs of bounded treedepth?
- Are there other structural parameters where can we obtain FPT algorithm? (eg. modularwidth)

Thank you for your attention!