25. Iterative Compression

Goal: Introduce iterative compression as an FPT-technique useful for minimization purposes.

Background: Became prominent with

EDGE BIPARTIZATION:

Given: Graph $G=(V,E)$.

Parameter: $k \in \mathbb{N}$

Question: Can we delete $\leq k$ edges to make G bipartite?

We illustrate iterative compression on VERTEXCOVER.

Observation: For minimization problems like VERTEXCOVER the following iterative compression is enough:

> Algorithm: Given a size $k+1$ solution to the minimization problem
> ↳ the algorithm either gives a certificate that there is no size k solution
> ↳ or it produces a solution of size k.

The algorithm is now used as follows:

↳ Iteratively add the vertices of the graph, one at a time.

↳ If the problem is compressible, we have
- a running solution of size $\leq k$ for the current subgraph
- or at some stage a certificate that there is no solution for G.

↳ The algorithm modifies the current solution at each (compression) stage to make one for the next stage.

Correctness: Crucially, iterative compression needs the property to be <u>hereditary</u> in the following sense:

If G has the property, then every induced subgraph has the property.

Algorithm (Iterative compression applied to VERTEXCOVER):

1.) Initially, set $C_0 := \emptyset$, $V_0 := \emptyset$.

2.) Until $V = V_s$ or the algorithm has returned "no",

set $V_{s+1} := V_s \cup \{v\}$ with $v \in V \setminus V_s$.

The induction hypothesis is

C_s is a vertex cover of $G|_{V_s}$ of size $\leq k$.

Consider $G|_{V_{s+1}}$.

3.) $C := C_s \cup \{v\}$ is a size $\leq k+1$ vertex cover of $G|_{V_{s+1}}$.

If C has size $\leq k$, go to next step.

If C has size $= k+1$, try to <u>compress</u>.

Consider all possible partitions $C = D \uplus Q$.

The idea is to <u>discard</u> the vertices in D

and <u>keep</u> the vertices in Q in the size k vertex cover.

The part D will be replaced by some H

which we construct.

For each such partition $D \uplus Q$:

↳ If there is an edge $\{u, v\} \subseteq D$ (and hence outside $Q \uplus H$),

then there is no vertex cover of $G|_{V_{s+1}}$ that avoids D.

$\rightarrow$ Otherwise, H must cover all edges that have

- one vertex in D and • the other one outside Q.

So if $\{u,v\}$ is not yet covered by $Q \uplus H$,
add the vertex $w \in \{u,v\} \setminus D$ to H.

4.) If, at the end, $H \uplus Q$ has size $\leq k$, set $C_{s+1} := H \uplus Q$.
Goto $s := s+2$.
Whenever $H \uplus Q$ exceeds size k, move to a new partition.

5.) If we tried all partitions and failed to move to step $s := s+2$,
return "no".

There are at most 2^{k+1} partitions to make in every step,
so the runtime is $O(2^k \cdot |G|)$. $\square$