Kernelization

A technique to design FPT algorithms

Filip Rydzi

Seminar on Algorithmics

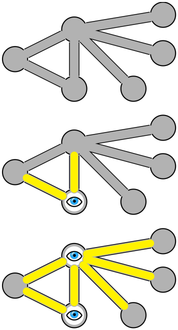
May 3, 2017

Motivation

Vertex cover

Given a graph G=(V, E), parameter k. We are asked to find a subset $C \subseteq V$, $|C| \leq k$, s.t. C touches all edges in E.

Figure: Finding vertex cover



Motivation[PS14]

- Simple algorithm to find Vertex Cover
  - $O(n^k)$ - enumerates all subsets of V (of size at most k) and checks if it's a valid vertex cover
  - $O(2^k \cdot n)$ - FPT algorithm
  - Which one is better ?
    Let's assume n=1000, k=15. The first algorithm needs 10^{45} steps, while the FPT only ca. $3 \cdot 10^7$

Fixed Parameter Tractable [PS14]

- Set of problems, which can be solved in $O(f(k) \cdot n^{O(1)})$ time
- Complexity class FPT
- FPT $\subseteq$ NP
- Techniques to design FPT algorithms
  - Depth-bounded search trees
  - **Kernelization**
  - Iterative compression
  - Treewidth
  - ...

Kernelization [LWG14]

- preprocessing procedure

Definition

Let π be a parametrized decision problem and (I,k) an instance of π. An algorithm A is called a kernelization algorithm for π iff

1. A transforms (I,k) to reduced instance (I',k') in polynomial time
2. (I,k) is a positive instance of π iff (I',k') is a positive instance of π
3. $|I'| \leq g(k)$ and $k' \leq k$

If the conditions above are satisfied, then we call π **kernelizable** and reduced instance (I',k') a **kernel**. If additionally g(k) is a polynomial function on k, then we say π admits a polynomial kernel.

A problem π is FPT $\equiv$ π admits a kernel

- π admits a kernel $\implies$ π is FPT [LWG14]
  Assume π admits a kernel, we reduced instance (I,k) to (I',k') in time $|I|^{O(1)}$.
  Now we can use an algorithm to solve (I',k') in time $O(f(k'))$.
  We obtained a solution in time $|I|^{O(1)} + O(f(k')) \leq O(f(k) \cdot n^{O(1)})$
  Thus π is FPT.

A problem π is FPT $\equiv$ π admits a kernel

- π is FPT $\implies$ π admits a kernel [Kim16]
  I.e. π can be solved by an FPT-algorithm A in time $O(f(k) \cdot n^c)$.
  Let's run A on an instance (I,k) for time n^{c+1}.
  - if A terminates and outputs YES/NO answer on (I,k), we produce constant-sized instance of π representing kernel.
  - if A doesn't terminate in time n^{c+1}, it means that $n < f(k) \implies$ (I,k) is already a kernel.

Kernelization on Vertex Cover

Input

Given a graph G=(V, E) and parameter k.

Reduction rules by Buss [PS14]:

1. If a vertex v has no neighbours, then delete v.
2. If a vertex v has exactly one neighbour u, then delete u and set k'=k-1.
3. If a vertex v has more than k neighbours, then delete v and set k'=k-1.

Kernelization on Vertex Cover

We need to show that the rules by Buss describe a valid kernelization preprocessing procedure on Vertex Cover.

1. We obtain a reduced instance (G',k') ✓
2. The positives of an instance is preserved (all reduction rules are safe).
   - If a vertex v has no neighbours, then delete v ✓
     Clearly a safe rule, there is no reason to include v in Vertex Cover.
   - If a vertex v has exactly one neighbour u, then delete u and set k'=k-1 ✓
     Safe rule, because we must include u in Vertex Cover to touch edge (u,v).

Kernelization on Vertex Cover

We need to show that the rules by Buss describe a valid kernelization preprocessing procedure on Vertex Cover.

1. We obtain a reduced instance (G',k') ✓
2. The positives of an instance is preserved (all reduction rules are safe).
   - If a vertex v has more than k neighbours, then delete v and set k'=k-1 ✓

     Safe rule too since we must include v to Vertex Cover, otherwise we would need to take at least k+1 vertices to Vertex Cover.

Kernelization on Vertex Cover

We need to show that the rules by Buss describe a valid kernelization preprocessing procedure on Vertex Cover.

1. We obtain a reduced instance (G',k') ✓
2. The positives of an instance is preserved (all reduction rules are safe) ✓

Kernelization on Vertex Cover

We need to show that the rules by Buss describe a valid kernelization preprocessing procedure on Vertex Cover.

3. $|G'| \leq g(k)$ and $k' \leq k$

   After some time we reach a point when no reduction rule by Buss can be applied further.

   This implies that each vertex has degree at most k.

   We distinguish the following cases:

   1. $|E(G)| \leq k^2$: (G,k) admits a polynomial kernel
   2. $|E(G)| > k^2$: (G,k) is a negative instance of Vertex Cover.

Kernelization on Vertex Cover

We need to show that the rules by Buss describe a valid kernelization preprocessing procedure on Vertex Cover.

1. We obtain a reduced instance (G',k') ✓
2. The positives of an instance is preserved (all reduction rules are safe) ✓
3. $|G'| \leq g(k)$ and $k' \leq k$ ✓

The rules by Buss describe a valid preprocessing procedure.

Other Kernelizable Problems

Planar Independent Set

Given a planar graph G=(V, E), parameter k. We are asked to find a subset $I \subseteq V$, $|I| \geq k$, s.t. no two vertices in I are adjacent in the original G.

In planar graphs there exists a vertex v with d(v) $\leq$ 5.

Planar Independent Set - kernelizable ?

In planar graphs there exists a vertex v with d(v) $\leq$ 5.
For any instance of k-Planar Independent set we can always find that vertex v with d(v) $\leq$ 5 and put either v or one of its neighbours in I. Then we decrement k and remove the chosen vertex and all of its neighbours from G and continue the procedure.
Until:

- we have no vertices left and $k > 0$ - no solution
- k=0 - positive instance!

Planar Independent Set - kernelizable ✓

We can formulate reduction rules:

- $|V| > 6k$ - positive instance
- $|V| \leq 6k$ - **we have a kernel** ✓

Planar Independent Set is a kernelizable problem. [Soc13]

Independent Set - kernelizable ?

Independent Set

Given a graph G=(V, E), parameter k. We are asked to find a subset $I \subseteq V$, $|I| \geq k$, s.t. no two vertices in I are adjacent in the original G.

Independent Set is not FPT and thus doesn't admit a kernel.

Another Kernelizable Problem

Edge Clique Cover

Given a graph G=(V, E), parameter k. We are asked to find cliques $C_1,\ldots,C_k \leq$ G such that $\bigcup_{i=1}^{k} E(C_i) = E(G)$.

Formulate reduction rules [GGHN09]:

1. Remove isolated vertices.
2. If there is an isolated complete graph K_2, we delete it and decrement k.
3. If there are vertices u, v connected by an edge and $N(u) = N(v)$, then delete u or v.

Edge Clique Cover

Formulate reduction rules [GGHN09]:

1. Remove isolated vertices.
2. If there is an isolated complete graph K_2, we delete it and decrement k.
3. If there are vertices u, v connected by an edge and $N(u) = N(v)$, then delete u or v.
   - Is this a safe rule ?

Edge Clique Cover

Formulate reduction rules [GGHN09]:

1. Remove isolated vertices.
2. If there is an isolated complete graph K_2, we delete it and decrement k.
3. If there are vertices u, v connected by an edge and $N(u) = N(v)$, then delete u or v.
   - Is this a safe rule ?
   - Yes, since any solution can be extended by adding the deleted vertex to the cliques containing its twin.

Edge Clique Cover

Formulate reduction rules [GGHN09]:

1. Remove isolated vertices.
2. If there is an isolated complete graph K_2, we delete it and decrement k.
3. If there are vertices u, v connected by an edge and $N(u) = N(v)$, then delete u or v.
4. If we cannot apply rule 1) to 3).
   - If $|V| > 2^k$ left - negative instance
     - Having more than 2^k vertices would mean that 2 vertices are in the same sets of cliques, i.e. they are true twins.

Edge Clique Cover

Formulate reduction rules [GGHN09]:

1. Remove isolated vertices.
2. If there is an isolated complete graph K_2, we delete it and decrement k.
3. If there are vertices u, v connected by an edge and $N(u) = N(v)$, then delete u or v.
4. If we cannot apply rule 1) to 3).
    - If $|V| > 2^k$ left - negative instance
    - else $|V| \leq 2^k$ - we have a **kernel**

Conclusion

For any instance of FPT problem we can apply the kernelization pre-processing algorithm to obtain one of the following results [AK10]:

- An early detection of negative instance
- An early detection of positive instance
- An equivalent instance whose size is bounded by a function of the parameter k

References I

- Faisal N. Abu-Khzam, *A kernelization algorithm for d-hitting set*, Journal of Computer and System Sciences **76** (2010), no. 7, 524 – 531.
- Jens Gramm, Jiong Guo, Falk Hüffner, and Rolf Niedermeier, *Data reduction and exact algorithms for clique cover*, J. Exp. Algorithmics **13** (2009), 2:2.2–2:2.15.
- Eunjung Kim.
- Y. Liu, J. Wang, and J. Guo, *An overview of kernelization algorithms for graph modification problems*, Tsinghua Science and Technology **19** (2014), no. 4, 346–357.
- A. G. A. Prasad and S. Shine, *Techniques for designing fixed parameter algorithms*, 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies (ICCICCT), July 2014, pp. 327–330.

References II

Arkadiusz Socala, *Introduction. sunflower lemma.*

References I

- 1 Vertex cover figure