Résolution exacte de problèmes NP-difficiles
Lecture 2: Kernelization

14 January, 2021 Lecturer: Eunjung Kim

1 Kernelization 101

A *kernelization* of P is a polynomial-time (in $|x|$ and k) algorithm which transforms an instance (x, k) of P into another instance (x', k') of P satisfying

- equivalence: $(x, k) \in P$ if and only if $(x', k') \in P$
- size bound: $|x| \leq g(k)$ and $k' \leq g(k)$ for some computable function

An instance (x', k') obtained after applying kernelization is called a kernel. The function $g(k)$ is the size of a kernel.

Usually, a kernelization consists in applying a sequence of *reduction rules*. A reduction rule for P is a polynomial-time (in $|x|$ and k) algorithm which transforms an instance (x, k) of P into an equivalent instance (x', k') of P. The equivalence of (x, k) and (x', k') is also referred to as the *soundness* or *safeness* of the reduction rule. Notice that the size of the resulting instance is not necessarily bounded. We say that an instance (x, k) is irreducible with respect to a reduction rule R is R cannot be applied to (x, k) anymore (or equivalently, applying R does not change the instance).

Lemma 1. *A decidable parameterized problem P is FPT if and only if it admits a kernelization.*

Proof: ($\Leftarrow$) If P admits a kernel of size $g(k)$, then run the kernelization algorithm (takes polynomial time in $|x| + k$) and then do an exhaustive search on the obtained kernel to decide whether $(x', k') \in P$. The whole procedure is an FPT-algorithm.
($\Rightarrow$) Let P has an FPT-algorithm $\mathcal{A}$ running in time $f(k) \cdot |x|^c$ for some constant c. Then run $\mathcal{A}$ on the given instance (x, k) for time $|x|^{c+1}$. If it ouputs YES/NO answer, then produce a constant-size instance of P accordingly. This would be the kernel. If $\mathcal{A}$ does not terminate in time $|x|^{c+1}$, this means $|x| < f(k)$. That is, the given instance (x, k) is already a kernel, and thus output (x, k). □

Devising a kernelization with small size bound $g(k)$ (usually, polynomial g) is one of the most active research topic in parameterized complexity. Kernelization design involve the following steps.

- Devise reduction rules.
- Prove that the above reduction rules are safe.
- Prove that when an instance (x', k') of P is irreducible w.r.t the reduction rules, $|x'| \leq g(k)$ and $k' \leq g(k)$. The smaller the function g is, the better. Notice that the equivalence of kernelization is automatically guaranteed by the safeness of reduction rules.

2 Simple kernelization for VERTEX COVER

We look at a simple kernelization for VERTEX COVER yielding $O(k^2)$ vertices.
Reduction Rule 1: If a vertex v is isolated in G, then delete v. The new parameter is $k' := k$.
Reduction Rule 2: If a vertex v is incident with at least $k+1$ edges in G, then delete v and set $k' := k-1$.

It is trivial to see that Reduction Rule 1 is safe. To see that Reduction Rule 2 is safe, notice that any vertex cover of size at most k in G must contain v. Hence, if (G,k) is a yes-instance, $(G-v,k-1)$ is also yes-instance. The opposite direction of equivalence is straightforward.

Consider an instance (G',k') for which none of Reduction Rules 1 and 2 can be applied, and analyze the size of G'. Since the parameter does not increase with the reduction rules, we know that $k' \leq k$.

Suppose G' is a yes-instance, and C is a vertex cover of G' with $|C| \leq k$. Since (G',k') is irreducible with respect to Reduction Rule 2, every $v \in C$ is incident with at most k edges of G. Therefore, there are at most $|C| \cdot k \leq k^2$ edges in G. Due to Reduction Rule 1, there's no isolated vertex in G' and thus every vertex in $V(G) \setminus C$ is adjacent with some vertex in C. As $deg(v) \in k$ for all $v \in C$, we have $|V(G)| = |V(G)) \setminus C| + |C| \leq |C| \cdot k + k = k(k+1)$.

Hence, if $|V(G')| > k(k+1)$ or $|E(G')| > k^2$, we know that (G',k') is a no-instance and output a constant-size no-instance as a kernel. Otherwise, (G',k') is a kernel.

3 LP-based kernelization for VERTEX COVER

Theorem 1 (Nemhauser-Trotter Theorem, NT Theorem in short)**.** *Given a graph G, a partition (R,H,C) satisfying the following can be computed in polynomial time.*

(a.) For any vertex cover S_r of $G[R]$, $S_r \cup H$ is a vertex cover of G.

(b.) There exists an optimal vertex cover containing H.

(c.) Any vertex cover of $G[R]$ is of size at least $\frac{1}{2}|R|$.

Before presenting an algorithm for computing such a partition (R,H,C), let's think how to use the above NT Theorem for computing a kernel. We propose the following reduction rules.

Reduction Rule 0: Remove isolated vertices from G.

Reduction Rule 1: Let (R,H,C) be a partition such that (a)-(c) of NT Theorem is met and $H \cup C \neq \emptyset$. Then delete $H \cup C$ from G and set $k' := k - |H|$, i.e. the new instance is $(G[R], k-|H|)$.

Lemma 2. *Reduction Rule 1 is safe.*

Proof: Suppose (G,k) is a yes-instance and let S be an optimal vertex cover. Notice that $|S| \leq k$. By condition (b) of NT Theorem, we can assume that $H \subseteq S$. Take $S_r := S \cap R$ and observe that S_r is a vertex cover of $G[R]$. Due to condition (a), $S_r \cup H$ is a vertex cover of G. Since $S_r \cup H \subseteq S$ and is a vertex cover of G, the optimality of S implies that $S \cap C = \emptyset$. Hence, $|S_r| = |S| - |H| \leq k - |H|$ and $(G[R], k - |H|)$ is a yes-instance.

For the opposite direction, suppose $(G[R], k - |H|)$ is a yes-instance and let S_r be a vertex cover of $G[R]$ of size at most $k - |H|$. By condition (a), $S_r \cup H$ is a vertex cover of G and its size is at most k. That is, (G, k) is a yes-instance. □

Lemma 3. VERTEX COVER *admits a kernel containing at most* $2k$ *vertices.*

Proof: Consider the following algorithm: find a partition (R, H, C) as in NT Theorem in polynomial time and apply Reduction Rule 1. (Reduction Rule 0 has been already applied). Let $(G', k') = (G[R], k - |H|)$ be the resulting instance.

- If $|R| > 2k$, then any vertex cover of G' contains more than k vertices by condition (c) of NT Theorem, and thus (G', k') is a no-instance. In this case, we output a constant-size no-instance.
- Otherwise $|R| \leq 2k$. We output (G', k').

In both cases, we output an instance equivalent to the initial instance containing at most $2k$ vertices. The running time follows from NT Theorem. □

How can we find a partition (R, H, C) as in NT Theorem? Now we consider this question, which leads to a proof of NT Theorem. There are several nice proofs of NT Theorem. Here we consider a version using Linear Programming formulation. Consider a Linear Programming Relaxation of VERTEX COVER .

$$
\begin{aligned}
\min \sum_{u \in V(G)} x_u & \\
x_u + x_v \geq 1 & \qquad \forall (u, v) \in E(G) \\
x_u \geq 0 & \qquad \forall u \in V(G)
\end{aligned}
$$

Observe that if an optimal solution to the above LP is integral, then it corresponds to an optimal vertex cover. In general, an optimal solution x^* to LP is not necessarily integral. We partition $V(G)$ according to their values in x^*.

- $R_0 := \{u \in V(G) : x^*_u = 0.5\}$
- $H_0 := \{u \in V(G) : x^*_u > 0,5\}$
- $C_0 := \{u \in V(G) : x^*_u < 0,5\}$

It is known that LP can be solved in polynomial time, hence the partition (R_0, H_0, C_0) can be found in polynomial time. We claim that this partition actually meets the condition (a)-(c) of NT Theorem. For this, we need the following lemma.

Lemma 4. *For every subset* $H'_0 \subseteq H_0$*, we have* $|H'_0| \leq |N(H'_0) \cap C_0|$.

Proof: Take $\epsilon = \min\{x^*_u - 0.5 : u \in H'_0\}$ and note that $\epsilon > 0$. Consider a solution x' defined as:

$$x'_u = \begin{cases} x^*_u - \epsilon & \text{if } u \in H'_0 \\ x^*_u + \epsilon & \text{if } u \in N(H'_0) \cap C_0 \\ x^*_u & \text{otherwise} \end{cases}$$

It is easy to verify that x' is a feasible LP solution. The objective value of x' equals

$$\sum_{u \in V(G)} x^*_u + \epsilon(-|H'_0| + |N(H'_0) \cap C_0)|).$$

From the optimality of x^*, our claim follows. □

Lemma 5. *The partition* (R_0, H_0, C_0) *meets the condition (a).*

Proof: Observe that C_0 is an independent set: indeed if there is an edge between $u, v \in C_0$, we have $x^*_u + x^*_v < 0.5 + 0.5 = 1$, violating the corresponding inequality in LP. For the same reason, there is no edge between C_0 and R_0. This means that $N(C_0) \subseteq H_0$, from which condition (a) holds. □

Lemma 6. *The partition* (R_0, H_0, C_0) *meets the condition (b).*

Proof: Let S be an optimal vertex cover of G and let S_r, S_h, S_c be its intersections with R_0, H_0 and C_0 respectively, and let $\bar{S}_h$ be the complement of S_h in H_0, that is, $\bar{S}_h := H_0 \setminus S_h$. We take a new set

$$S' := (S \setminus S_c) \cup \bar{S}_h,$$

i.e. the set obtained from S by removing all of C_0 and adding all of H_0. This is a vertex cover due to Lemma 5. We claim that S' is again an optimal vertex cover. To see this, it suffices to show that the vertices newly added are not more than those removed, that is, $|\bar{S}_h| \leq |S_c|$.

Observe that $N(\bar{S}_h) \cap C_0 \subseteq S_c$ since otherwise S_c fails to cover all edges incident with $\bar{S}_h$. Applying Lemma 4 for $H'_0 := \bar{S}_h$, we have

$$|\bar{S}_h| \leq |N(\bar{S}_h) \cap C_0| \leq |S_c|.$$

□

Lemma 7. *The partition* (R_0, H_0, C_0) *meets the condition (c).*

Proof: By Hall's theorem[1] and Lemma 4, there is a matching M between H_0 and C_0 saturating H_0. Let us fix such a matching M and notice that $|M| = |H_0|$. For any feasible solution x' to the above LP, its objective value is

$$\begin{aligned} \sum_{u \in V(G)} x'_u &\geq \sum_{u \in R_0} x'_u + \sum_{u \in H_0 \cup C_0} x'_u \geq \sum_{u \in R_0} 0.5 + \sum_{u \in V(M)} x'_u \\ &= \frac{1}{2}|R_0| + \sum_{(u,v) \in M} (x'_u + x'_v) \geq \frac{1}{2}|R_0| + |M| = \frac{1}{2}|R_0| + |H_0|. \qquad (\star) \end{aligned}$$

[1] Hall's theorem: Let G be a bipartite graph with vertex bipartition (X, Y). There is a matching M saturating X (i.e. every vertex of X is incident with an edge in M) if and only if $|X'| \leq |N(X')|$ for every $X' \subseteq X$.

Let S_r be an arbitrary vertex cover of $G[R_0]$. Since (R_0, H_0, C_0) meets condition (a) due to Lemma 5, $S_r \cup H_0$ is a vertex cover. Since an optimal LP solution x^* provides a lower bound for $S_r \cup H_0$, we have

$$|S_r| + |H_0| \geq \sum_{v \in V(G)} x_v^* \geq \frac{1}{2}|R_0| + |H_0|,$$

where the second inequality follows from the previous inequality ($\star$). This complete the proof. □