Lecture Outline 8
Topics in Graph Algorithms (CSCI5320-19S)

CAI Leizhen
Department of Computer Science and Engineering
The Chinese University of Hong Kong
lcai@cse.cuhk.edu.hk

March 7, 2019

Keywords: Kernelization

1. *Kernelization* formalizes the idea of efficient data reduction and is a useful tool for obtaining FPT algorithms. A kernelization is a polynomial algorithm that transforms each instance (I, k) of a parameterized problem to an equivalent instance (I', k') that satisfies (a) $|I'| \leq g(k)$ for some function $g(k)$ and (b) $k' \leq k$. We call the reduced instance (I', k') a *kernel*.

   When a parameterized problem has a kernel and is decidable, we can use exhaustive search on the kernel to solve the problem in $n^{O(1)} + f(k)$ time.

2. **Theorem**: *A parameterized problem is FPT iff it admits a kernel and is decidable.*

   Let $f(k)n^c$ be the running time of an FPT algorithm. If $f(k) \leq n^c$ then the algorithm runs in time n^{2c} which is polynomial time. Otherwise $n \leq f(k)^{1/c}$ and the instance itself is a kernel.

   However, kernel sizes for different FPT problems vary a lot, some are linear, some polynomial, and some exponential.

3. It seems difficult to formalize the idea of efficient data reduction under traditional complexity. For instance, reducing I to I' with $|I'| = |I| - 1$ in polynomial time would imply that I is solvable in polynomial time.

4. An easy way to obtain a kernel is to show that the problem is polynomial-time solvable when $|I| \geq g(k)$ for some function $g(k)$.

5. $4k$ kernel for Independent k-Set on planar graphs.

   By the 4-color theorem, an n-vertex planar graph G always contains an independent set of size $\geq k$ when $n \geq 4k$. Otherwise (G, k) is a kernel with less than $4k$ vertices.

6. $O(2k)$ kernel for $(n-k)$-Dominating Set.

   We use the following result of Ore: *A graph with no isolated vertices has a dominating set of size $\leq \lfloor n/2 \rfloor$.*

If $k < n/2$ then $n - k > n/2$, and the answer is always yes. Otherwise, $n \leq 2k$ and (G, k) is a kernel with $\leq 2k$ vertices.

7. $O(k^2)$ kernel for k-EDGE ODD SUBGRAPH (Cai and Yang 2010)

   Are there k edges E' in G such that every vertex of $G[E']$ has an odd degree?

   Easy if there is a vertex of degree $\geq k+1$. Otherwise G has a matching M with k edges if G has $\geq k(2k-1)$ edges, and M forms a solution. This implies a kernel with $O(k^2)$ vertices and edges.

   Open Problem: Does k-EDGE ODD SUBGRAPH admit a kernel with $O(k)$ vertices?

8. *Reduction rule*: Many kernelization algorithms are based on reduction rules.

9. k-VERTEX COVER: Kernel with k^2 vertices.

   Rule 1: If v is an isolated vertex, then delete v from G.

   Rule 2: If $d(v) = 1$, then add the neighbor u of v to the solution, delete u from G, and decrease k by 1.

   Rule 3: If $d(v) > k$, then add v to the solution, delete v from G, and decrease k by 1.

   We obtain a graph G' and a parameter k' after repeatedly applying the above 3 reduction rules. If G' has more than kk' edges then G has no k-vertex cover, else (G', k') is a kernel with $\leq k^2$ vertices and $\leq k^2$ edges.

10. k^2 kernel for COVERING POINTS BY LINES

    Are there $\leq k$ lines that cover all points in a given set S of points?

    Observation: It suffices to consider lines L passing through pairs of points in S.

    Rule 1: If a line $l \in L$ covers at least $k+1$ points, remove it from L, remove all points covered by l from S, and decrease k by 1.

    After repeatedly applying Rule 1, every line covers $\leq k$ points. No solution if S contains more than k^2 points, and otherwise we obtain a kernel with k^2 points.

11. $k(k+2)$ kernel for ARC REVERSAL FOR TOURNAMENT

    Is it possible to reverse $\leq k$ arcs in a tournament G (i.e., a digraph obtained from an orientation of a complete graph) to make it a dag?

    Rule 1: If arc e is contained in $\geq k+1$ triangles, reverse e and reduce k by 1.

    Rule 2: If a vertex v is not contained in any triangle, then delete v.

    Correctness for Rule 1 is obvious. For Rule 2, let X (resp., Y) be out-neighbors (resp. in-neighbors) of v. Then no arc goes from X to Y, and we reverse arcs in $G[X]$ and $G[Y]$ only. Since every arc is contained in at least one triangle and at most k triangles, the reduced Yes-instance contains at most $k(k+2)$ vertices.

12. $O(k^3)$ kernel for HITTING SET FOR TRIPLES.

    Given a collection C of triples from a ground set S, we want to determine whether there are $\leq k$ elements S' in S that hit every triple in C.

Rule 1: If a pair $\{x, y\}$ is contained in more then k triples, then delete all these triples and add $\{x, y\}$ to C.

Note that in this case, either x or y must be in the hitting set.

Apply Rule 1 until it no longer applies, and then apply the following Rule 2 repeatedly.

Rule 2: If an element z is contained in more than k^2 pairs and triples, then delete all these pairs and triples, add z to the hitting set and reduce k by 1.

Note that in this case, z must be in the hitting set.

Answer "No" if C has more than k^3 pairs and triples, and otherwise we have a $O(k^3)$ kernel.

13. $O(2^k)$ kernel for Edge Clique Cover (Gramm et. al. 2006)

    Does G contain $\leq k$ complete graphs (cliques) that cover all edges?

    Rule 1. Remove isolated vertices.

    Rule 2. If there is a component with a single edge, remove it and reduce k by 1.

    Rule 2. If there is an edge uv satisfying $N[u] = N[v]$, remove vertex u.

    Let G be a graph with more than 2^k vertices that has a clique cover $C_1, \ldots, C_k$. Assign each vertex v a binary vector x_v of length k where the i-th bit is 1 iff v is contained in clique C_i. Since there are only 2^k different vectors of length k, some pair u, v of vertices satisfy $x_u = x_v$. If $x_u = x_v = 0$ then Rule 1 applies. Otherwise u and v are contained in same cliques, implying that u and v are connected and share the same neighborhood, Rule 2 applies if uv forms a component, and otherwise Rule 3 applies.

14. $2k$ kernel for k-Vertx Cover

    For a graph $G = (V, E)$, the minimum vertex cover problem can be formulated as an ILP: $\min \sum_{v \in V} x_v$ subject to $x_u + x_v \geq 1$ for each edge and $x_v \in \{0, 1\}$ for each vertex.

    We can obtain an LP-relaxation as follows: $\min \sum_{v \in V} x_v$ subject to $x_u + x_v \geq 1$ for each edge and $0 \leq x_v \leq 1$ for each vertex.

    For an optimal LP solution of G, let

$$V_0 = \{v \in V : x_v < 1/2\},$$

$$V_{1/2} = \{v \in V : x_v = 1/2\},$$

and

$$V_1 = \{v \in V : x_v > 1/2\}.$$

Theorem (Nemhauser and Trotter 1975) Every graph G contains a minimum vertex cover V' satisfying $V_1 \subseteq V' \subseteq V_1 \bigcup V_{1/2}$.

Proof. Let $\{x_v : v \in V\}$ be an optimal LP solution of G. We prove that G contains a minimum vertex cover $V' \subseteq V_1 \bigcup V_{1/2}$ and leave the $V_1 \subseteq V'$ part as an exercise.

Let $A = V' \cap V_0$ and $B = V_1 - V'$, and set $\epsilon = \min\{x_v - 1/2 : v \in B\}$. Suppose $A \neq \emptyset$. If $|A| < |B|$, we can obtain a smaller LP solution by decreasing x_v by ϵ for each vertex $v \in B$ and increasing x_u by ϵ for each vertex $u \in A$. Therefore $|A| \geq |B|$ by the optimality of the LP solution, and $(V' - A) \cup B$ is a required vertex cover of G. ∎

The above theorem can be used to obtain a $2k$ kernel as follows: Let $G' = G[V_{1/2}]$ and $k' = k - |V_1|$. As the size of any vertex cover of G is at least $|V_{1/2}|/2$, G has no k-vertex cover if $|V_{1/2}| > 2k'$. Otherwise (G', k') is a kernel with $\leq 2k$ vertices.

Question: Is there an $\epsilon > 0$ so that k-VERTX COVER admits a kernel with at most $(2 - \epsilon)k$ vertices?