CMSC 858F: Algorithmic Lower Bounds: Fun with Hardness Proofs
Fall 2014
Fixed Parameter Algorithms and Lower Bounds for parameterized Problems

Instructor: Mohammad T. Hajiaghayi
Scribe: Aditya Acharya

October 21-23, 2014

1 Overview

In this lecture, we go over the basics of Fixed Parameter Tractable (FPT) problems. We define the related concept of Kernelization, introduce an appropriate notion for reduction in parameterized complexity, giving various examples along the way. We conclude with W-Hierarchy, which represents various complexity classes restricted to parameterized problems.

2 Fixed Parameter Tractability

Loosely speaking our algorithmic goals can be classified into three broad categories:

1. Obtaining *exact* and optimal solution
2. Computing the solution *fast*, i.e. in poly time.
3. Tackling *hard* problems, i.e. NP-hard problems

Unless $\mathsf{P} = \mathsf{NP}$ we have to satisfy ourselves with any two out of the three goals. Most of the early undergrad algorithms like matching, shortest path etc. are exact and fast. To tackle hard problems and obtain a fast solution we use approximation algorithms, PTAS etc. FPT or fixed parameter tractable algorithms come to our rescue when we need to tackle hard problems yet obtain an optimal solution. The idea behind fixed-parameter tractability is to take an

NP-hard problem, and to try to separate out the complexity into two pieces - some piece that is polynomial in the size of the input, and another piece that depends on some "parameter" to the problem. Many problems have the following form: given an object x of size n and a nonnegative integer k, does x have some property that depends on k? We call such a problem parameterized.

Definition 1 *A parameterized problem* $L\{n, k\}$ *is fixed-parameter tractable if it has a running time of* $f(k) \cdot n^{O(1)}$*, where* f *is an arbitrary function depending only on* k*. The corresponding complexity class is called* FPT*.*

Theorem 1 $k-\mathtt{vertexcover}$ *is in FPT*

Proof: We recall $k-\mathtt{vertexcover}$ is the problem of deciding whether a given graph $G \equiv (V, E)$ has a vertex cover of size k. We begin by observing the following simple fact.
Observation 1 *Let* $(u, v) \in E$ G *has a vertex cover of size* k *iff at least one of* $G-u$ *or* $G-v$ *has a vertex cover of size* $\leq k-1$
Let $VC(G, k)$ denote the boolean version of the $k-\mathtt{vertexcover}$ problem. From the above observation it follows $VC(G, k) = VC(G-u, k-1) \vee VC(G-v, k-1)$. We note the base cases: $VC(H, 0)$ is false for non empty H and true for empty H. All of the base cases and the operations $G-u$ can be done in $O(n)$ time. hence we can find $VC(G, K)$ in $2^k n$ time as there are k levels of recursion and each recursion takes $O(n)$ time. ∎
The best known vertex cover FPT algorithm runs in $O(1.2378^k + kn)$ [1].

FPT algorithm having a running time of $O\left(n^k\right)$ is generally considered bad, e.g. $k-\mathtt{clique}$ problem. Running time of $2^k n$ or in general $f(k)n$ is good for e.g. $k-\mathtt{vertexcover}$ problem. Having a sub-exponential running time in the parameter k is considered great for example, $k-\mathtt{vertexcover}$ problem on planar graphs has a $2^{O\left(\sqrt{k}\right)} n^{O(1)}$ time algorithm.

3 Kernelization

Definition 2 *Kenelization is the procedure of reducing a problem* $X\{n, k\}$ *down to* $X\{f(k), g(k)\}$ *using only* $n^{O(1)}$ *preprocessing time.*

We say a problem has a kernel if it can be kernelized. From the definition, having a kernel implies a running time of $n^{O(1)} + f(k)$. But in fact we know something much stronger.

Theorem 2 *A parameterized problem has a kernel if and only if it is fixed parameter tractable*

In general $f(k)$ can be exponential but a good kernel should be polynomial or even linear in k.

Next we look at a simple kernelization for the vertex cover problem.

- First add any vertex of degree more than k to the vertex cover and remove the edges incident upon them, as otherwise we would need at least $k+1$ vertices to cover edges incident on it.
- Now find a *maximal matching*(a set of edges H without common vertices which is maximal i.e. no edges can be added to H)

Note that the size of the maximal matching cannot be more than k otherwise we would need more than k vertices to cover these edges in the matching. Also each vertex has to be the neighbor of at least one of the vertex in the matching (by maximality). Now the maximum degree of any vertex in the matching is k. Hence the size of the kernel is $2k^2$ or $O(k^2)$. The currently best known kernelization algorithm in terms of the number of vertices is due to Lampis (2011) [2] and achieves $2k - c\log k$ vertices for any fixed constant c.

It is not possible, in this problem, to find a kernel of size $O(\log k)$, unless $P = NP$, for such a kernel would lead to a polynomial-time algorithm for the NP-hard vertex cover problem. However, much stronger bounds on the kernel size can be proven in this case: unless $\mathsf{coNP} \subseteq \mathsf{NP/poly}$ (believed to be unlikely), for every $\epsilon > 0$ it is impossible in polynomial time to find kernels with $O(k^{2-\epsilon})$ edges [3]. It is unknown for vertex cover whether kernels with $(2-\epsilon)k$ vertices for some $\epsilon > 0$ would have any unlikely complexity-theoretic consequences.

4 FPT hardness or parameterized Complexity

To build a complexity theory for parameterized problems, we need two things

- An appropriate notion of reduction (poly time reductions are not helpful)
- An appropriate hypothesis

To see why polynomial time reductions are not good we consider the following example. It is easy to see that a graph G has an independent set of size k (a set of k vertices with no edges between them) if and only if it has a vertex cover of size $n-k$. This transforms an independent set of instance (G, k) to a vertex cover problem of instance $(G, n-k)$ through a poly time reduction. However $vertex - cover$ is in FPT but $independent - set$ is not known to be in FPT.

4.1 parameterized reduction

Let (X, k) be an instance of parametric problem P. We consider ϕ as a parameterized reduction from P to Q if

- $\phi(X)$ can be computed in time $f(k) \cdot |X|^{O(1)}$
- $\phi(X)$ is a yes instance of $Q \iff X$ is a yes instance of P.
- If k is the parameter of the instance X and k' the parameter of instance $\phi(X)$ then $k' \leq g(k)$ for some function g.

Note that by the above definition if there is a parametric reduction from problem P to Q and Q is FPT then P is also FPT. Transforming $\mathsf{independent\_set}(G, k)$ to $\mathsf{vertex\_cover}(G, n-k)$ is not a parameterized reduction. But transforming $\mathsf{independent\_set}(G, k)$ to a $\mathsf{clique}(\bar{G}, k)$ is a parameterized reduction.

Before moving into example of paametrized reductions, we define a useful variant of clique called multicolored clique, which heps in simplifying details in FPT reductions by allowing an almost systematic gdget construction.

Definition 3 $\mathsf{multi\_colored\_clique}$*: Given* k *colors and a graph* G *whose vertices are colored with one of the* k *colors, does there exist a clique containing one vertex fom each color class?*

Note that multicolored versions exists for all natural subset problems e.g. independent set, and one can show that they are as hard as their uncolored counterparts [4]

Theorem 3 *There is a parameterized reduction from* $k-\mathsf{clique}$ *to* $\mathsf{multi\_colored\_clique}$

Proof: Given an instance (G, k) for $k-\mathsf{clique}$, we construct a graph G' by taking k copies $v_1, v_2, \cdots, v_k$ for each vertex v and color v_i with color i. We then add an edge in G' between two vertex u_i and v_j , $i \neq j$ iff u and v are connected in G'. It is straightforward to verify that G has a k-clique iff G' has a k-multicolored clique. ■

Theorem 4 *There is a parameterized reduction from* $\mathsf{multi\_colored\_independent\_set}$ *to* $k-\mathsf{dominating\_set}$

Proof: First we recall the definition of $k-\mathsf{dominating\_set}$: Given a graph G and a number k does there exist a set D of size k such that the rest of the $n-k$ vertices have at least one neighbor in D?

Let G be a graph with its vertices being k-colored. We construct a graph H such that G has a multicolored clique if and only if H has a dominating set of size k. Let $G, k, (V_1, \cdots, V_k)$ be an instance of the $\mathsf{multi\_colored\_independent\_set}$ with V_i being the set of vertices colored i. We construct a graph H as follows

- For every vertex $v \in V(G)$, we introduce v in H
- For every $1 \leq i \leq k$, we make the set V_i in H a clique by adding th required edges.
- For every $1 \leq i \leq k$, we introduce two new vertices x_i , y_i into H and make them adjacent to every vertex in V_i
- For every edge $e \in E(G)$ with endpoints $u \in V_i$ and $v \in V_j$, we introduce a vertex w_e into H and make it adjacent to every vertex in $(V_i \cup V_j) \setminus \{u, v\}$

We claim G has a k colored independent set iff H has a dominating set of size k.

Let $Z \subset V(G)$ be an independent set such that $Z \equiv \{z_1, z_2, \cdots, z_k\}$ and $z_i \in V_i$. Now we show Z is a dominating set in H. x_i and y_i are dominated by z_i. All of the vertices in V_i's are dominated by z_i. Now suppose $w_{\{u,v\}}$, with $u \in V_i$ and $v \in V_j$ does not have a neighbor in Z. That implies $u = z_i$ and $v = z_j$, which further implies that z_i and z_j have an edge in G [contradiction].

Next, let's assume we have a dominating set D of size k in H. Since all of the x_i's and y_i's have to be dominated each V_i must have at least one of its member in D. Hence we can write D as $D = \{d_1, d_2, \cdots, d_k\}$ such that $d_i \in V_i$. Now we show that D is a k-independent set of G. Of course all of the vertices in D are uniquely colored in G as they belong to different V_is. Now suppose that d_i and d_j have an edge between them in G. That implies $w_{\{d_i,d_j\}}$ does not have a neighbor in D [contradiction]

■

Indeed lots of other parameterized problems are known to be as hard as Clique such as Set cover, Hitting set, Connected Dominating set, Independent dominating set, partial vertex cover etc.

Exercise 1 *Prove that there is a parameterized reduction from Dominating set to Connected dominating set and Set cover.*

So far we have been relying on just $\mathsf{P} \neq \mathsf{NP}$ for parameterized complexity. Often we need stronger assumptions like

- *[Clique hardness]:* $k - \mathsf{clique}$ or $k - \mathsf{independent\_set}$ cannot be solved in $f(k)n^{O(1)}$ time
- *[Exponential Time Hypothesis]:* n-variable $3 - \mathsf{SAT}$ cannot be solved in $2^{O(n)}$ time.

The second assumption is much stronger and in fact implies the first [5].

4.2 W-Hierarchy

From the previous subsection we have a parameterized reduction from Independent Set to Dominating Set problem, but what about the reverse? The aswer most probably is 'No'. Unlike NP completeness where most problems are equivalent, here we have a hierarchy of hardness. Let us first define few important concepts

Definition 4 *Boolean circuit is a circuit consisting of input gates, negation ($\sim$), AND ($\bigwedge$), OR ($\bigvee$) and output gates.*

Definition 5 *Circuit Satisfiability: Given a Boolean circuit C, decide if there is an assignment on the inputs of C such that the output is true*

Figure 1 shows an example of Circuit SAT

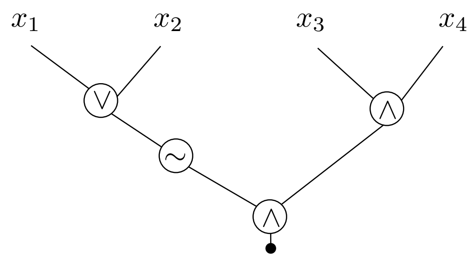

Figure 1: Circuit SAT

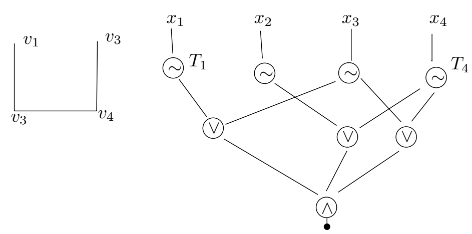

Figure 2: Graph G and its corresponding WCSAT for Independent set problem

Definition 6 *Weighted Circuit satisfiability: Given a Boolean circuit C and an integer k, decide if there is an assignment of weight k such that the output is true, where the weight is the number of true inputs.*

Definition 7 *The depth of a circuit is the maximum length of a path from an input to the output*

Definition 8 *The weft of circuit is the maximum number of large gates on a path from an input to the output, where large gates are the gates that have more than two inputs.*

Next we briefly show the reduction of independent set to a weighted circuit satisfiability. Figure 2 shows such an example. For every vertex ν_i in G we create a corresponding literal x_i. For every literal x_i we negate it by attaching it to a negation gate T_i. For every edge $e = \{\nu_i, \nu_j\}$ we OR T_i and T_j. Finally we AND all the outputs from the OR gates to get the final output. It is easy to see that such a construction of a weighted circuit SAT is euivalent to the independent set problem. We also observe that weft of such a circuit is 1 and depth is 3.

Similarly we can show that Dominating set reduces to weighted circuit SAT. Figure 3 gives an example. For every vertex ν_i in G we create a corresponding literal x_i. For every literal x_i we create an OR gate T_i which has as input x_j iff $\nu_j \in \nu_i \cup \mathsf{N}(\nu_i)$, where $\mathsf{N}(\nu)$ is the set of neighbors of ν. Next we AND all the outputs from T_is. The depth and weft both equal two for such a circuit.

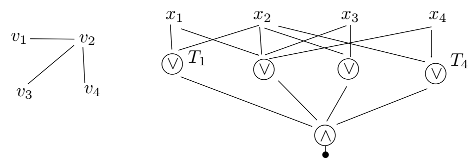

Figure 3: Graph G and its corresponding WCSAT for Dominating set problem

Let $C[t, d]$ be the set of all circuits having weft at most t and depth at most d.

Definition 9 *A problem P is in class $W[t]$ if there is a constant d (that holds for all instances of the problem and may be a function of k) and a parameterized reduction from P to weighted circuit SAT of $C[d, t]$*

Hence Independent set problem is in $W[1]$ and the dominating set problem is in $W[2]$. Moreover it can be proven that Independent-Set and Dominating Set problem are $W[1]$ and $W[2]$-complete respectively. Note that FPT algorithms are in $C[O(f(k)), 0]$ and thus in $W[0]$.

Also $W[i] \subseteq W[j]$ for all $i \leq j$. Hence if there is a parameterized reduction from Dominating set to Independent set, $W[1] = W[2]$, on account of those problems being complete in their respective classes. Under parameterized reductions it is believed the $\subseteq$ is indeed $\subset$.

Many natural computational problems occupy the lower levels $W[1]$, $W[2]$. XP is the class of parameterized algorithms that can be solved in $n^{f(k)}$ time. So to conclude we have $\mathsf{FPT} = W[0] \subseteq W[1] \subseteq W[2] \subseteq \cdots \subseteq \mathsf{XP}$

References

[1] Jianer Chen, Iyad A. Kanj, and Ge Xia. 2010. Improved upper bounds for vertex cover. Theor. Comput. Sci. 411, 40-42 (September 2010), 3736-3756. DOI=10.1016/j.tcs.2010.06.026 http://dx.doi.org/10.1016/j.tcs.2010.06.026

[2] Lampis, Michael (2011), A kernel of order 2k - c log k for vertex cover, Information Processing Letters (Elsevier) 111 (2324): 10891091, doi:10.1016/j.ipl.2011.09.003

[3] Dell, Holger; van Melkebeek, Dieter (2010), "Satisfiability allows no non-trivial sparsification unless the polynomial-time hierarchy collapses", STOC 2010, pp. 251260

[4] Fellows, Michael R; Hermelin, Danny; Rosamond, Frances; Vialette, Stphane,On the parameterized complexity of multiple-interval graph problems,Theoretical Computer Science,410,1,53-61,2009,Elsevier

[5] Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, Ge Xia, Strong computational lower bounds via parameterized complexity, Journal of Computer and System Sciences, Volume 72, Issue 8, December 2006, Pages 1346-1367, ISSN 0022-0000, http://dx.doi.org/10.1016/j.jcss.2006.04.007