CMSC 858F: Algorithmic Lower Bounds
Fall 2014
Parameterized reductions and assumptions

Instructor: Mohammad T. Hajiaghayi
Scribe: Sina Dehghani

October 28, 2014

1 Overview

In the previous lecture we concluded FPT $= W[0] \subseteq W[1] \subseteq W[2] \subseteq \ldots \subseteq \mathsf{XP}$. In summary, by parameterized reductions we can show lots of problems are at least as hard as clique or dominating set ans thus in $W[1]$ or $W[2]$. In this lecture, we study exponential time hypothesis and several hardness results in planar graphs.

2 Exponential time hypothesis (ETH)

(introduced by Impagliazzo, Paturi, Zane [2])

Definition 1 *ETH: there is no $2^{o(n)}$ time algorithm for n variable 3-SAT. (the current best bound is 1.30704^n [1]).*

Note that an n-variable 3-SAT can have $\Omega(n^3)$ clauses but Impagliazzo et al. [2] show that there is a $2^{o(n)}$-time algorithm for n-variable 3-SAT *iff* there is a $2^{o(m)}$-time algorithm for m-clause 3-SAT. Thus ETH also says: There is no $2^{o(m)}$-time algorithm for m-clause 3-SAT.

The standard textbook NP-hardness reduction from 3-SAT to 3-coloring constructs a graph of $O(n+m) = O(m)$ edges and $O(n+m) = O(m)$ vertices to solve 3-SAT instance of n-variables and m-clauses. Thus assuming ETH, there is no $2^{o(n)}$ algorithm for 3-coloring on an n-vertex graph G.

Since there are many standard polynomial-time reductions from 3-coloring to many other problems such that the reduction increases the number of vertices by at most a constant factor, we have

Corollary 1 *Assuming ETH, there is no* $2^{o(n)}$*-time algorithm on vertex graphs for Independent set, Clique, Dominating set, Vertex cover, Hamiltonian path, Feedback vertex set, etc..*

Similarly, since in the problems above $k \leq n$ we have no $2^{o(k)}n^{O(1)}$ algorithm for the k version of the problems above. (Note that $2^{o(n)}n^{O(1)} = 2^{o(n)}$).

2.1 Tighter Bounds

As aforementioned ETH implies Clique hardness; indeed we can prove a much stronger and more interesting theorem

Theorem 1 (chen et al.'04 [3]) *: Assuming ETH, there is no* $f(k)n^{o(k)}$ *algorithm for* k*-clique for any computable function* f *which is increasing.*

Proof: Assume by contradiction k-clique can be solved in time $f(k)n^{\frac{k}{n^{s(k)}}}$, where $s(k)$ is a monotone increasing unbounded function. We use this algorithm to solve 3-coloring on an n-vertex graph G in time $2^{o(n)}$. Let k be the largest integer such that $f(k) \leq n$ and $k^{\frac{k}{s(k)}} \leq n$. Thus function $k := k(n)$ is monotone increasing and unbounded. Now we split the vertices of G into k groups. Let us build a graph H where each vertex corresponds to a proper 3-coloring of one of the groups and connect two vertices if they are not conflicting. Thus every k-clique of H corresponds to a proper 3-coloring of G and thus a 3-coloring of G can be found in time $f(k)|V(H)|^{\frac{k}{s(k)}}$. Since $f(k) \leq n$ and the partition into k groups

$$f(k)|V(H)|^{\frac{k}{s(k)}} \leq n \cdot (k3^{\frac{n}{k}})^{\frac{k}{s(k)}} = n \cdot k^{\frac{k}{s(k)}} \cdot 3^{\frac{n}{s(k)}} \leq n^2 \cdot 3^{\frac{n}{s(k)}}.$$

Since $k := k(n)$ and $s(k)$ are monotone increasing and unbounded, $n^2 \cdot 3^{\frac{n}{s(k)}} = 2^{o(n)}$. ■

It is easy to see that if we have a reduction from k-clique instance (x, k) to an instance $(x', g(k))$ of problem A, then $f(k)n^{o(g^{-1}(k)}$ algorithm for A implies $f(k)n^{o(k)}$ for k-clique. Thus

- To rule out $f(k)n^{o(k)}$ algorithms for A, we need a parameterized reduction that blows up the parameter at most linearly.
- To rule out $f(k)n^{o(\sqrt{k})}$ algorithms for A, we need a parameterized reduction that blows up the parameter at most quadratically.
- Thus assuming ETH, there is no $f(k)n^{o(k)}$ algorithm for Set cover, Hitting set, Connected dominating set, Independent dominating set, Partial vertex cover, and Dominating set in bipartite graphs.

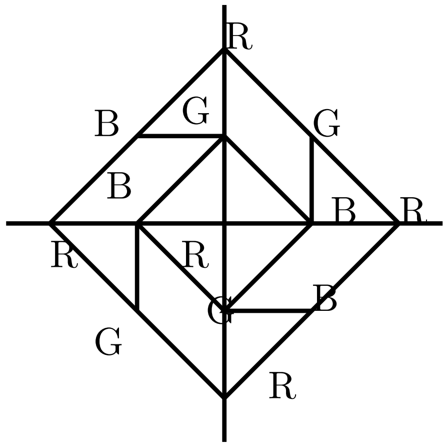

Figure 1:

2.2 Planar graphs

There is a standard (NP-complete) reduction from 3-coloring in general graphs to planar 3-coloring which uses a "cross over gadget" below.

Claim 1 *In every 3-coloring of the gadget, opposite external connectors have the same color.*

Claim 2 *Every coloring of the external connectors where the opposite vertices have the same color can be extended to the whole gadget.*

Thus if two edges cross, we replace them with a cross over gadget and Claims 1 and 2 guarantee that two end-points of an edge have different colors.

The reduction from 3-coloring to planar 3-coloring introduces $O(1)$ new edges/vertices. Thus a graph with m edges and n vertices can be drawn with $O(m^2)$ crossings.

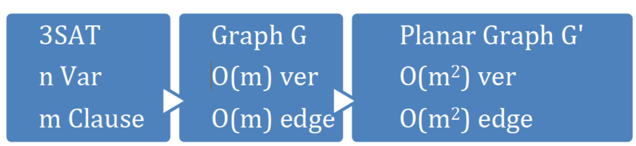

Figure 2:

Corollary 2 *Assuming ETH, there is no* $2^{o(\sqrt{(n)})}$ *algorithm for 3-coloring on* n*-vertex planar graph* G *(observed by [Cai and Juedes [4]]).*

consequence: Assuming ETH, there is no $2^{o(\sqrt{(n)})}$-time and no $2^{o(\sqrt{(k)})}n^{O(1)}$-time algorithm on planar graphs for (k)-Independent, (k)-Dominating set, (k)-Vertex cover, (k)-path, (k)-Feedback vertex set, etc.

Note that above bounds are indeed **tight** due to bidimensionality algorithms.

To get stronger and more bounds, Grid tiling is the key problem (introduced by [Marx [5]])

Grid tiling: Input: a $k \times k$ matrix and a set of pairs $S_{i,j} \subseteq [D] \times [D]$ for each cell.

Output: find a pair $s_{i,j} \in S_{i,j}$ for each cell such that

- Vertical neighbors agree in the first coordinate.
- Horizontal neighbors agree in the second coordinate.

(1, 1) (3, 1) $\underline{(2, 4)}$	(5, 1) $\underline{(1, 4)}$ (5, 3)	(1, 1) $\underline{(2, 4)}$
$\underline{(2, 2)}$ (1, 4)	(3, 1) $\underline{(1, 2)}$	$\underline{(2, 2)}$
$\underline{(2, 3)}$ (1, 3)	(1, 1) $\underline{(1, 3)}$	$\underline{(2, 3)}$

Figure 3:

Theorem 2 *Grid tiling is $W[1]$-hard.*

Proof: We use a reduction from k-clique. Given a graph G for k-clique we construct a $k \times k$ grid as follows. For each cell (i, j):

- For $i = j$, the pair $(x, y) \in S_{i,j}$ *iff* $x = y$.
- For $i \neq j$, the pair $(x, y) \in S_{i,j}$ *iff* x and y are adjacent in G.

Now each diagonal cell defines a vertex in clique. ■

Indeed the above reduction gives a stronger bound.

Theorem 3 $k \times k$ *Grid tiling is $w[1]$-hard and assuming ETH cannot be solved in time $f(k)n^{o(k)}$ for any function f.*

This lower bound is the key for proving hardness results for planar graphs or even general graphs (the matrix is like a grid which is planar).

Examples

- List coloring on planar graphs

- Multiway cut on planar graphs with k terminals
- Independent set for unit disks
- (*)Planar directed Steiner forest with k terminals

Let's see an overview of the proof of (*).

We want to show DSF(Directed Steiner forest which connects k terminal pairs (s_i, t_i) in a directed graph), under ETH does not have any algorithm $f(k)n^{o(k)}$ even on planar DAGs.

Here we have $2k$ terminals (a_i, b_i) and (c_i, d_i) as shown in Figure 4. We construct the graph in Figure 4 which has the same size as Grid tiling and thus $f(k)n^{o(k)}$ hardness of Grid tiling gives the same hardness for DSF even on planar DAGs (combine hardness of Theorem 4, 5 with hardness of Grid tiling - see more details in [Chitnis, Hajiaghayi, Marx [6]]).

If $(x, y) = s_{i,j} \in S_{i,j}$ then we color green the vertex in the gadget $G_{i,j}$.

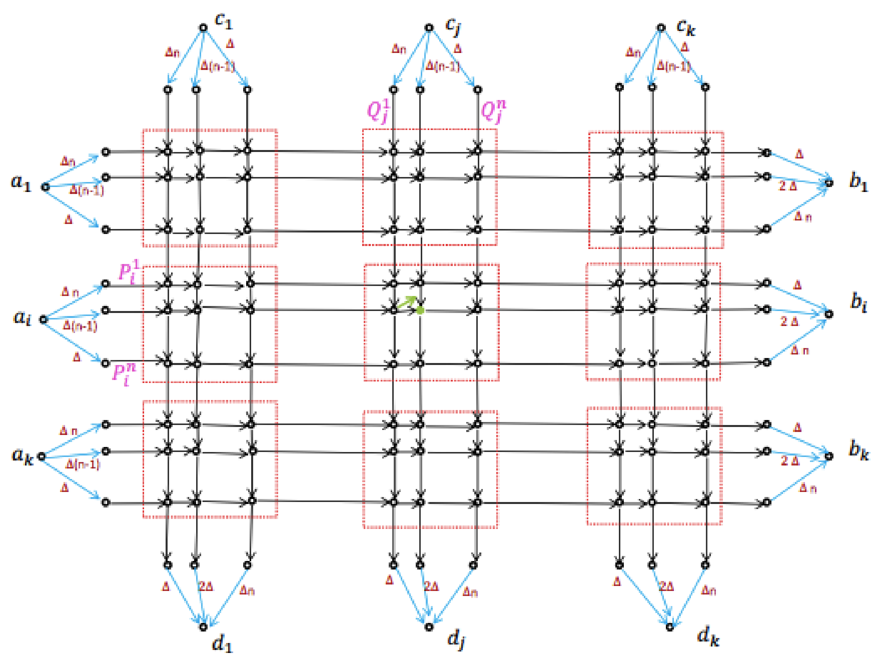

Figure 4:

We need a small technical modification: we add one dummy row and column to the Grid tiling instance. Essentially we now have a dummy index 1. So neither the first row nor the first column of any $S_{i,j}$ has any elements in the Grid tiling instance. That is, no green vertex can be in the first row or the first column of any gadget. Combining this fact with the oriental of the edges we get the only

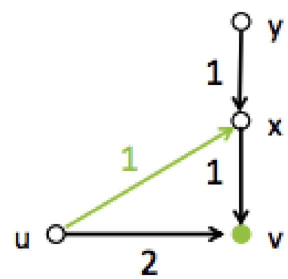

Figure 5: Let u, v be two consecutive vertices on the canonical path say P_i^l. Let v on the canonical path $Q_l^{l'}$ and let y be the vertex preceding it on this path. If v is a green vertex then we subdivide the edge (y, v) by introducing a new vertex x and adding two edges (y, x) and (x, v) of weight 1. We also add an edge (u, x) of weight 1. The idea is if both the edges (y, v) and (u, v) were being used initially then now we can save a weight of 1 by making the horizontal path choose (u, x) and then we get (x, v) for free, as it is already being used by the vertical canonical path.

gadgets which can intersect any $a_i \rightsquigarrow b_i$ path and $G_{i,1}, G_{i,2}, \ldots, G_{i,k}$. Similarly the only gadgets which can intersect any $c_j \rightsquigarrow d_j$ paths are $G_{1,j}, G_{2,j}, \ldots, G_{k,j}$.

We now prove two theorems which together give a reduction from Grid tiling to DSF.

Theorem 4 *Grid tiling has a solution implies OPT for DSF is at most* $\beta - k^2$.

Proof: For each $1 \leq i, j \leq k$ let $s_{i,j} \in S_{i,j}$ be the vertex in the solution of the Grid tiling instance. Therefore for every $i \in k$ we know that each of the k vertices $s_{i,1}, s_{i,2}, \ldots, s_{i,k}$ have same x-coordinate, say α_i. Similarly for each $j \in [k]$ each of the k vertices $s_{1,j}, s_{2,j}, \ldots, s_{k,j}$ has the same y-coordinate, say λ_j. We now use the canonical path $P_i^{\alpha_i}$ for (a_i, b_i) and the canonical path $Q_j^{\lambda_j}$ for (c_i, d_i). Each of the $c_j \rightsquigarrow d_j$ paths will pay the full weight of a canonical path, which is $\frac{\beta}{2k} = \Delta(n+1) + (2k+1) + 2k(n-1)$. However each $a_i \rightsquigarrow b_i$ path will encounter a green vertex in each of the k gadgets along the way and save k in each path. Hence over all terminals we save a weight of k^2 and we have a solution to DSF of weight $\beta - k^2$. ■

Theorem 5 *Optimum for DSF is at most* $\beta - k^2$ *implies Grid tiling has a solution.*

Proof: The proof is more involved and uses the concept of canonical path. ■
Finally note that one probably most important result in lower bound for kernelization is that:

Theorem 6 *For any $\epsilon > 0$, the vertex cover problem parameterized by solution size does not admit a polynomial kernel with bitsize (in this case the number of the edges in $O(k^{2-\epsilon})$ unless $NP \subseteq CoNP/poly$.*

Though we do not prove theorem above since it is involved, we can use parameterized reduction to get hardness for other problems (since we can always pipeline the parameterized reduction into kernelization algorithm). e.g. consider Feedback vertex set (deleting a minimum the number of vertices to make a graph cycle-free).

There is a simple parameterized transformation that takes an instance (G, k) of vertex cover and outputs an equivalent instance (G', k) of Feedback vertex set. Take every edge $uv \in E(G)$ and add a vertex w_{uv} that is adjacent only to u and v (in G'); thus creating a triangle that must hit by the solution. Thus we get a similar theorem to Theorem 6 for Feedback vertex set as well.

We have more involved theorems as well.

Theorem 7 *Let $\epsilon > 0$, be any constant. Unless $NP \subseteq CoNP/poly$;*

- *For any $q \geq 3$, the q-SAT problem parameterized by the number of variables n does not have a kernel of bitsize $O(n^{q-\epsilon})$.*
- *For any $d \geq 2$, the d-hitting set problem parameterized by solution size k does not have a kernel with bitsize $O(k^{d-\epsilon})$. (In hitting set we want to find a subset of minimum size that intersects every set of collections C which has $\mathsf{size} \leq d$)*

Note that theorem above says trivial kernelization which just removes duplicates of clauses or sets is the best that we can do.

Another important theorem ia as follows for Steiner tree (which asks for minimum the number of edges which connect a terminal set $T \subseteq V(G)$.)

Theorem 8 *Steiner tree parameterized by the size of the tree does not admit a polynomial kernel unless $NP \subseteq CoNP/poly$.*

Corollary 3 *The same holds if Steiner tree is parameterized by $|T|$ instead of solution size (since $|T| \leq \mathsf{solutionsize} + 1$).*

3 Fixed parameter algorithms and other fields

3.1 FPT and Approximation

An FPT optimum approximation algorithm for a problem O with approximation ratio ρ is an algorithm A that, given an input x output a $y \in \mathsf{sol}(x)$ such that

$$\begin{cases} \mathsf{cost}(x, y) \leq \mathsf{opt}(x)\rho(\mathsf{opt}(x)) & \text{if goal is min.} \\ \mathsf{cost}(x, y) \geq \mathsf{opt}(x)/\rho(\mathsf{opt}(x)) & \text{if goal is max.} \end{cases}$$

We require that on input x that algorithm A runs in $f(\mathsf{opt}(x))|x|^{O(1)}$ time for some computable function f. Chitnis, Hajiahayi, and Kortsaz [8] prove

Theorem 9 *Under ETH and another conjecture projection Games Conjecture(PGC), there exists constants $F_1, F_2 > 0$ such that the set cover problem does not admit an FPT optimum approximation algorithm with ratio $\rho(\mathsf{opt}) = \mathsf{opt}^{F_1}$ in $2^{\mathsf{opt}^{F_2}}\mathsf{poly}(N, M)$ time where N is the size of the universe and M is the number of sets.*

Theorem 10 *Unless $\mathsf{NP} \subseteq \mathsf{SUBEXP}$, for every $0 < \delta < 1$, there exists a constant $F(\delta) > 0$ such that clique has no FPT optimum approximation with ratio $\rho(\mathsf{opt}) = \mathsf{opt}^{1-\delta}$ in $2^{\mathsf{opt}^F}\mathsf{poly}(n)$ time where n is the number of vertices in the graph.*

There are a few $f(\mathsf{opt})$ approximation for some W-Hard problems but still the field is very new we expect more results to be known in the field.

3.2 FPT and Streaming

Though in streaming and semi-streaming, we often assume the space needed is a function of n. However for many reasonable graph problems, we have can assume the solution is not large on real-world input instances. Thus we can aim for parameterized streaming in which we require the memory be in $\widetilde{O}(k) = O(k\ \mathsf{polylog}\ n)$.

Chitnis, Carmode, Hajiaghayi, and Monemizadeh [7] introduce the concept of parameterized streaming and obtain such algorithms for matching and vertex cover. They consider two models

1. insertion only in which edges will be just added.
2. dynamic streaming in which we have both insertion and deletion of edges.

[7] obtain results for both cases. However for dynamic streaming they only consider the case that the solution size is at most k during the entire course of the stream. This area is novel and new and obtaining new hardness results for it is very interesting, e.g., if the size of the solution just at the end of the stream is at most k and in the middle can be large.

References

[1] Hertli, Timon. "3-SAT Faster and Simpler—Unique-SAT Bounds for PPSZ Hold in General." SIAM Journal on Computing 43.2 (2014): 718-729.

[2] Impagliazzo, Russell, Ramamohan Paturi, and Francis Zane. "Which problems have strongly exponential complexity?." Foundations of Computer Science, 1998. Proceedings. 39th Annual Symposium on. IEEE, 1998.

[3] Chen, Jianer, et al. "Tight lower bounds for certain parameterized NP-hard problems." Computational Complexity, 2004. Proceedings. 19th IEEE Annual Conference on. IEEE, 2004.

[4] Cai, Liming, and David Juedes. "Subexponential parameterized algorithms collapse the W-hierarchy." Automata, languages and programming. Springer Berlin Heidelberg, 2001. 273-284.

[5] Marx, Daniel. "On the optimality of planar and geometric approximation schemes." Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on. IEEE, 2007.

[6] Chitnis, Rajesh, MohammadTaghi Hajiaghayi, and Dniel Marx. "Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset." SIAM Journal on Computing 42.4 (2013): 1674-1696.

[7] Chitnis, Rajesh, et al. "Parameterized Streaming Algorithms for Vertex Cover." arXiv preprint arXiv:1405.0093 (2014).

[8] Chitnis, Rajesh, MohammadTaghi Hajiaghayi, and Guy Kortsarz. "Fixed-parameter and approximation algorithms: A new look." Parameterized and Exact Computation. Springer International Publishing, 2013. 110-122.