FPT hardness for Clique and Set Cover with super exponential time in k

Mohammad T. Hajiaghayi * Rohit Khandekar† Guy Kortsarz‡

Abstract

We give FPT-hardness for SETCOVER and CLIQUE with super exponential time in the parameter k. Under the $\eth$ and PGC we prove that SETCOVER admits no $\log^{1+c} k$ ratio, for $c > 0$ for any algorithm with running $t(k) = \exp\left(k^{(\log k)^f)}\right) \cdot \text{poly}(n)$ for constant $f > 0$. Under the ETH alone, we prove that SETCOVER admits no $\sqrt{\log k}$ ratio, in time $\exp\left(k^{(\log^f k)}\right) \cdot \text{poly}(n)$ for constant $f > 0$.

Under the ETH we prove CLIQUE admits no $(c, t(k))$, approximation, for any constant c in time $exp(exp(k^d))$ with d a constant that depends on c.

1 Introduction

1.1 Basic notation

When using reductions from 3-SAT to optimization problems P, the number of variables is always denoted by q, the number of clauses is always denoted by m. The size of the instance I of P is always denoted by n.

A function $f : \{1, 2, \ldots\} \mapsto \{1, 2, \ldots\}$ is *proper* if it is a computable, non decreasing function, We deal with proper function only and so we do not mention that again.

*Maryland University. Supported in part by NSF CAREER award 1053605, NSF grant CCF-1161626, ONR YIP award N000141110662, and DARPA/AFOSR grant FA9550-12-1-0423. Email: hajiagha@cs.umd.edu.

†KCG holdings Inc., USA Email: rkhandekar@gmail.com

‡Computer Science department, Rutgers University, Camden. Email: guyk@crab.rutgers.edu Partially supported by NSF grant number 1218620.

We denote by poly(n) an arbitrary polynomial in n and by polylog(n) an arbitrary polynomial in $\log n$.

2 Background

2.1 FPT-hardness

For a minimization (resp, maximization) problem P, of size n and a parameter k, an algorithm is called an $(r(k), t(k))$-FPT-approximation algorithm for P if for any instance I of P with optimum OPT, the algorithm either computes a feasible solution for I, with value at most $k \cdot r(k)$ (resp, at least $k/r(k)$) or it computes a certificate that $k < \text{OPT}$ (resp, $k > \text{OPT}$), in time $t(k) \cdot \text{poly}(n)$. For maximization problems the property $k/r(k) = o(k)$ is required as well. A problem P is (r,t)-FPT-inapproximable (or, (r,t)-FPT-hard) if the problem does not admit an (r,t)-FPT-approximation algorithm.

3 The Exponential Time Hypothesis

The following is our main assumption. It is due to Impagliazzo, Paturi and Zane. [7] and to the the sparsification lemma of [1], by Calabro, Impagliazzo and Paturi.

The ETH Conjecture: The 3-SAT problem admits no $2^{o(q+m)}$ exact solution, with with q the number of variables in the instance, and m the number of clauses.

The *Ga-ETH* conjecture states that for any constant $\epsilon > 0$ the 3-SAT admits no $2^{o(m+q)}$ time algorithm that if a given 3-SAT formula, is fully satisfiable, or only $1 - \epsilon$ fraction of the clauses can be simultaneously satisfied.

3.1 Setcover and Clique

A CLIQUE is a set of vertices all two pairs of which are neighbors.

CLIQUE
Input: an undirected graph $G(V, E)$, of size n, and an integer k representing the optimum value.

Question: Find in time $t(k) \cdot \text{poly}(n)$ if there exists CLIQUE of size at least k. or produce a proof that $k > \text{OPT}$.

In the SETCOVER problem the input is a universe U and a collection $\mathcal{S}$ of subset $\{S_1, S_2, \ldots\}$ of $\mathcal{S}$. A set cover is a subset $\mathcal{S}'$ of $\mathcal{S}$ so that the union of the sets in $\mathcal{S}'$ contains all of U.

SETCOVER

Input: U, $\mathcal{S}$ of size n and a parameter k representing the optimum value

Question: Find in time $t(k) \cdot \text{poly}(n)$ if there is a SETCOVER of size at most k, or prove that $k < \text{OPT}$.

3.2 The $W[i]$ hierarchy

A minimization (respectively, maximization) problem belongs to FPT if given an instance I of the problem of size n and a parameter k, for some function t there is an algorithms that tells if the optimum is k or less (respectively, k or more) in time $t(k) \cdot \text{poly}(n)$ or otherwise reruns a certificate that $k < \text{OPT}$ (respectively, $k > \text{OPT}$). The W-hierarchy is a collection of computational complexity classes. We will not define the $W[i]$ classes, but rather speak on the implication of being $W[i]$-hard for $i \geq 1$. It is well known that $\text{FPT} = W[0] \subseteq W[1] \subseteq W[2] \subseteq W[3], \ldots$.

It is widely believed that $\text{FPT} \neq W[i]$ for any $i \geq 1$. Therefore $W[i]$-hard problems for $i \geq 1$, are widely believed to be fixed parameter intractable. The CLIQUE problem is $W[1]$-hard and the SETCOVER problem is $W[2]$-hard. Hence we should not expect FPT algorithm for these problems. We can hope that there may be FPT-approximation for these problems. Our paper and others cast serious doubt even on this easier task.

3.3 The Fellows conjecture

The following is a conjecture by Fellows.

- SETCOVER is (r, t)-FPT-hard for any functions r and t.
- CLIQUE is (r, t)-FPT-hard for any functions r and t.

We are not aware of a citation for this conjecture.

In [2], (see FOCS 2016), it is stated that *"its hard to use the PCP to prove fixed parameter hardness, as the value of* OPT *in these gap reductions is usually very large"*. However, three years earlier [] we used the PCP to prove FPT-hardness for both SETCOVER and CLIQUE with time that is super exponential in k.

4 Related work

In [3], hardness results for CLIQUE and SETCOVER with sub-exponential time $t(opt)$ are studied.

Downey et. al [5], proved that there is no constant additive approximation for the the parameterized SETCOVER problem. The Minimum Maximal independent set (MMIS) problem is given $G(V, E)$ of size n and a parameter k, find in time $t(k) \cdot \text{poly}(n)$ if there is an independent set of size at most k that is also a dominating set, or prove that $k < \text{OPT}$. In [5] an $(r(k), t(k))$-hardness is, given for MMIS for any two functions r and t. In [11], it is shown that a problem called weighted monotone/antimonotone circuit satisfiability does not admit an (r, t) approximation for any two functions r, t.

The most related paper to our paper proves:

Theorem 4.1. [2] **FOCS 2016**
For any function t, SETCOVER *admits no $\log^{1/4} k$ ratio algorithm that runs in time $t(k)$.*

As we shall see, this result is incomparable to our results for SETCOVER.

5 Our results

From now on $exp(t)$ means in this paper c^t for some constant $c > 1$. In the next statement we cite the projection game conjecture (PGC) that **is defined only later.** However, the implications can be understood **without knowing the conjecture**.

Theorem 5.1. *Under the* ETH *and* PGC *conjectures,* SETCOVER *is (r, t)-FPT-hard for $r(k) = (\log k)^\gamma$, for a constant $\gamma > 1$, in time $t(k) = \exp(\exp((\log k)^\gamma)) \cdot \text{poly}(n) = \exp\left(k^{(\log^f k)}\right) \cdot \text{poly}(n)$ and $f = \gamma - 1$.*

As indicated above the time here is much larger than just exponential in k. Further $\Omega(\log k)$ hardness follows trivially from the known $\Omega(\log n)$ inapproximability (under $P \neq NP$) for SETCOVER [15]. We prove a stronger hardness as $\gamma > 1$.

A stronger version of the PGC that assumes a size $m \cdot P_1(\log\log m)$ PCP (Moshkovitz. Private communication) implies the following:

Theorem 5.2. *Under* ETH *and the above stronger version of* PGC *there exists constants $d_1, d_2 > 0$ so that* SETCOVER *is (r, t)-FPT-hard for $r(k) = k^{d_1}$ and $t(k) = \exp(\exp(k^{d_2}))$.*

Under the ETH alone we get:

Theorem 5.3. *Under* ETH *alone,* SETCOVER *cannot be approximated within $c\sqrt{\log k}$ for* *some constant c, in time* $\exp\left(k^{(\log k)^f}\right) \cdot \text{poly}(n)$ *for some constant* f.

Theorem 5.3 gives a $\sqrt{opt}$ hardness which is stronger than $\log^{1/4} k$ hardness of [2]. On the other hand [2] proves their hardness for any running time $t(k)$, while our running time is some *specific* super exponential in k, running time $t(k)$. Thus, Theorem 5.3 and [2], are *incomparable*. However, our paper gives a lower bound for CLIQUE and [2] does not.

Hardness for clique: Marx [10] raised the question if there is some function t so that CLIQUE can be approximated by 2 in time $t(OPT)$. We prove that to get an approximation of 2 for CLIQUE, t must be doubly exponential in k or larger, partially answering the question of Marx.

Theorem 5.4. *Under the* ETH, *for any constant* c, CLIQUE *is* (c, t)-FPT-*hard for* $t(k) = \exp(\exp(k^d))$ *for some constant* d *that depends on* c. *It is also* (r, t)-*fpt-hard for some super constant function* $r(k)$ *and* $t(k) = \exp(\exp(k/p(k)))$ *for an arbitrarily slowly increasing function* $p(k)$.

6 A well known reduction from 3-SAT to setcover

The following theorem is well known from the standard reduction of PCP to Labelcover and the reduction from Labelcover to SETCOVER. When the size of the PCP is $m \cdot 2^{\log^{\alpha} m} \cdot P_1(\log m)$ we get:

Theorem 6.1. *There exists a polynomial* $P_3 > P_1$ *and a reduction from* SAT *to* SETCOVER *with* $m \cdot 2^{\log^{\alpha} m} \cdot P_3(\log m)$ *sets and with* $O(m^5)$ *elements so that in a yes instance the optimum equals* $m \cdot 2^{\log^{\alpha}} m \cdot P_1(\log m)$ *and in the no instance* $opt_n \geq opt_y \cdot d \cdot \log_2 n$. *for some constant* d

7 FPT-hardness for Set Cover with super-exponential time in k

We start with the SETCOVER instance of Theorem 6.1. For simplicity we assume that all terms that follow are integral. Correcting the proof using $\lfloor$ and $\rfloor$ is trivial. The idea is to make the optimum much smaller preserving the gap. We introduce a set $s \in \mathcal{S}'$ as $s = \cup_{i=1}^{p} s_i$ for each subcollection $\{s_1, s_2, \ldots, s_p\} \subseteq \mathcal{S}$ of size $p = m/\log m$.

Lemma 7.1. *The number of sets in the new instance* $S' = (U, \mathcal{S}')$ *is* $2^{o(m)}$. *The new instance can be constructed in time* $2^{o(m)}$.

Proof. Recall that the number of sets is $n = m \cdot 2^{\log^\alpha m} \cdot P_3((\log(m)))$. The new instance size is at most

$$\binom{n}{p} = \binom{m \cdot 2^{\log^\alpha m} \cdot P_1((\log(m)))}{m/\log m}.$$

We use the inequality $\binom{n}{p} \leq (ne/p)^p$ to upper-bound this by

$$\left(e \cdot 2^{\log^\alpha m} \cdot P_1(\log m) \cdot 2\log m\right)^{m/\log m} = 2^{O(m/\log^{1-\alpha} m)} = 2^{o(m)}.$$

The last equality holds since $0 < \alpha < 1$. It is easy to see that the new instance can be created in $2^{o(m)}$ time. □

The new optimum for a yes instance, $k = opt_y/(m/\log m) = k_y(s) = opt_y \cdot \log m/m$ and of a no instance $k_n(s) = opt_n \cdot \log m/m$. Thus the gap $opt_y(s)/opt_y(s)$ remains at least $d \cdot \log n$. And the optimum $k = k_y(s)$ is $k = opt_y(m) = 2^{\log^\alpha m} \cdot \log m \cdot P_1((\log(m)))$.

Define $r(k) = (\log k)^\gamma = \exp\left(k^{(\log^f k)}\right) \cdot \text{poly}(s)$ and $t(k) = \exp(\exp((\log k)^\gamma))$ for any $1 < \gamma < 1/\alpha$, as given in Theorem 5.1. Note that $r(k) = \tilde{O}((\log^\alpha m)^\gamma) = o((\log^\alpha m)^{1/\alpha}) = o(\log m)$ and $t(k) = 2^{o(m)}$. Thus, SETCOVER is $(r(k), t(k))$-FPT-hard for these functions, proving Theorem 5.1.

7.1 Stronger hardness for setcover via a stronger assumption

In this section we prove Theorem 5.2 We assume

Conjecture 7.2. *There exists a constant $c > 0$ and a* PCP *of size $m \cdot \text{poly}\log(m)P_1(1/\epsilon)$, for any ϵ so that $\epsilon \geq 1/m^c$.*

This above conjectured is due to Moshkovitz (private communication). By Corollary 6.1, and the above conjecture we get the following corollary, using $\epsilon = c'/\log^2 m$ for a large enough constant c':

Corollary 7.3. *There exists a a reduction from 3-*SAT *to* SETCOVER *so that:*

1. *The number of sets is $\sigma = m \cdot P_1(\log(m))$*
2. *The number of elements is* $\text{poly}(m)$.
3. *The value of the optimum in yes instance is* exactly $opt_y = m \cdot P_2((\log(m)))$ *and that in the "no" instance is at least* $\log m \cdot opt_y$.

A proof of along the lines of the proof of Theorems 5.1 and 5.4 gives that SETCOVER is $(k^{d'}, exp(exp(k^{d_2})))$-FPT-hard. for some constant d_2.

7.2 Hardness under the ETH alone

The following is proved in [13]. Let $P_1(m)$ and $P_2(m)$ be two polynomials as in Corollary 6.1.

Theorem 7.4. *There exists a constant c and a* PCP *of size $m \cdot 2^{\log^\alpha m} \cdot \mathrm{poly}(1/\epsilon)$, such that the size of the alphabet is at most $\exp(1/\epsilon)$ and the gap that can be chosen to be $1/\epsilon$ for any $\epsilon > 1/m^c$.*

Proof. We choose $\epsilon = \ln 2 \cdot \log^\alpha m$. Note that $exp(\epsilon) = 2^{\log^\alpha m}$. Therefore when using the reduction from 3-SAT to SETCOVER described in Corollary 6.1 we get:

Corollary 7.5. *There exists a constant $d > 0$, and a constant $0 < \alpha < 1$ and a reduction from 3-*SAT *to* SETCOVER *so that:*

1. *The number of sets is $m \cdot 2^{2\log^\alpha m}$*
2. *The number of elements is $\mathrm{poly}(m)$.*
3. *The gap is $d \cdot \sqrt{\log^\alpha m}$.*
4. *The optimum of a yes instance does not change, namely, is $k_y(s) = 2^{\log^\alpha m} \cdot \mathrm{poly}\log(m)$.*

Note that the optimum does not change because we still select one vertex out of every supervertex.

The gap is $d\sqrt{\log^\alpha k}$ for some constant d. $k = k_y(s) = 2^{\log^\alpha k}$. Thus for some constant c, the problem is $(c \cdot \sqrt{\log k}, \exp\left(k^{(\log^f k)}\right))$-*fpt* hard for the same constant $f > 0$, that appears in Theorem 5.1. □

8 Fixed parameter hardness for clique

In this section we present the inapproximability for CLIQUE. Note that *graph products* play a crucial role in the proof.

Theorem 8.1. [4] *There exist a reduction from* SAT *of size m to a 3-*SAT *of size $S \leq m \cdot \mathrm{poly}(m)$, so that for a yes instance all clauses of the* 3-SAT *can be satisfied, and for a no instance at most $1/2$ of the clauses can be simultaneously satisfied*

Theorem 8.2. [6] *There exists a constant ϵ and linear reduction from* 3-SAT *to* CLIQUE*, of some size S_1 so that for a yes instance $opt_y = m'$ and for a no instance $k_n \leq (1-\epsilon)m'$ and the number of vertices is $7 \cdot m'$*

Clearly $S_1 = 7m' = O(m \cdot \text{poly}(m))$ because the reduction is linear.

Composing the reductions:

Corollary 8.3. *There exists a reduction from the decision version of* 3-SAT *to the optimization version of* CLIQUE, *of size* S_1, *so that the number of vertices in the* CLIQUE *instance is* $S_1 = O(m \cdot \text{poly}(\log m))$ *so that* $k_y/opt_n \geq 1/(1-\epsilon)$ *for some constant* ϵ. *The optimum* $opt_y = m'$ *for a yes instance is known.*

Proof of the theorem 5.4 We create a new CLIQUE instance and later set k to be the new optimum value $k = opt_y(s)$. Let $f(m)$ be any super constant slowly growing function. Introduce a vertex into the new graph for each subset of size $S_1/(b \cdot \log\log m \cdot f(m))$ vertices in the old CLIQUE instance, with b a constant to be defined. Every such set is called a supervertex. Two supervertices A, B, are connected by an edge, if $A \cup B$ is a clique, *and* $A \cap B = \emptyset$. The last condition, namely, the fact that two sets that are connected must be disjoint is not needed in the SETCOVER reduction, but it is crucial here.

The constant b is such that $\text{poly}(\log m) \leq (\log m)^b$ for the term $\text{poly}(\log m)$ In Corollary 8.3.

Lemma 8.4. *The new instance of the* CLIQUE *problem has size* $2^{o(m)}$.

Proof. The size of the original clique is bounded by $S_1 = O(\cdot m \cdot \log^b m)$. We take all subsets of size $S_1/(b \cdot \log\log m \cdot f(m))$. Using $\binom{n}{p} \leq (ne/p)^p$, we get that the number of supervertices is at most $O(\log^b m \cdot m)^{m/((b \log\log m) f(m))} = 2^{O(m)/f(m)} = 2^{o(m)}$. The last inequality is because $f(m) = \omega(1)$. The number of edges in the new CLIQUE instance, being at most the square of the number of vertices, is also $2^{o(m)}$. □

Lemma 8.5. *The gap between a yes and a no instance remains* $1/(1-\epsilon)$

Proof. Let $\rho = S_1/(b \cdot \log\log m \cdot f(m))$. Clearly, $opy_y(s) = opt_y/\rho$ and $opt_n(s) = opt_n/\rho$. Hence the ratio between $opt_n(s)$ and $opt_y(s)$ does not change and is at least $1/(1-\epsilon)$. □

The resulting hardness: We got a constant gap in which the optimum (of a yes instance) is $k = opt_y(s) = O(\log m^b)$. Thus $k^d = o(\log m)$ for some constant d. Thus $t(k) = \exp(\exp(k^d)) = 2^{o(m)}$. Hence the problem admits a $(c, \exp(\exp(k^d)))$-FPT-hardness

Graph products: Graph products with parameter i, take a graph of size n and produce a graph of size n^i so that the gap grows to $(1/(1-\epsilon))^i$ and $k = k_y(s)$ grows to $k = opt_y(s)^i$, Taking a constant graph product, we can make $r(k) \geq c$ for any constant c. The running time will still $\exp(\exp(k^d)))$ except that d is a constant that depends on c.

To get super constant $r(k)$ we use graph products with $i = \sqrt{(f(m))}$ and the size of the graph remains $2^{o(m)}$ by Claim 8.4. Thus gives a gap of $r(k) = (1/(1-\epsilon))^{\sqrt{f(m)}}$. Clearly $r(k)$ is a super constant function. The increase in the size of the optimum mean that $t(k) = \exp(\exp(k/p(k)))$ for an arbitrarily slowly increasing function $p(k)$.

References

[1] C. Calabro, R. Impagliazzo, and R. Paturi. A duality between clause width and clause density for sat. In *CCC*, pages 252–260, 2006.

[2] Y. Chen and B. Lin. The constant inapproximability of the parameterized dominating set problem. *FOCS 2016, to appear*, 2016.

[3] R. H. Chitnis, M. Hajiaghayi, and G. Kortsarz. Fixed-parameter and approximation algorithms: A new look. In *Parameterized and Exact Computation - 8th International Symposium, IPEC 2013, Sophia Antipolis, France, September 4-6, 2013, Revised Selected Papers*, pages 110–122, 2013.

[4] I. Dinur and S. Safra. The importance of being biased. In *STOC*, pages 33–42, 2002.

[5] R. G. Downey, M. R. Fellows, C. McCartin, and F. A. Rosamond. Parameterized approximation of dominating set problems. *Inf. Process. Lett.*, 109(1):68–70, 2008.

[6] U. Feige, S. Goldwasser, L. Lovász, S. Safra, and M. Szegedy. Interactive proofs and the hardness of approximating cliques. *J. ACM*, 43(2):268–292, 1996.

[7] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? In *FOCS*, pages 653–663, 1998.

[8] G. Kortsarz. On the hardness of approximating spanners. *Algorithmica*, 30(3):432–450, 2001.

[9] C. Lund and M. Yannakakis. On the hardness of approximating minimization problems. *J. ACM*, 41(5):960–981, 1994.

[10] D. Marx. Parameterized complexity and approximation algorithms. *Comput. J.*, 51(1):60–78, 2008.

[11] D. Marx. Completely inapproximable monotone and antimonotone parameterized problems. *J. Comput. Syst. Sci.*, 79(1):144–151, 2013.

[12] D. Moshkovitz. The projection games conjecture and the NP-hardness of $\ln n$-approximating set-cover. In *APPROX*, pages 276–287, 2012.

[13] D. Moshkovitz and R. Raz. Two-query PCP with subconstant error. *J. ACM*, 57(5), 2010.

[14] M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 182–191, 1995.

[15] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *STOC*, pages 475–484, 1997.