Laboratoire d'Analyses et Modélisation de Systèmes pour l'Aide à la Décision
UMR 7243

CAHIER DU LAMSADE

321

Mai 2012

Subexponential and FTP-time Inapproximability of Independent Set and Related Problems

B. Escoffier, E.J. Kim, V.Th. Paschos

Subexponential and FPT-time Inapproximability of Independent Set and Related Problems*

Bruno Escoffier, Eun Jung Kim, and Vangelis Th. Paschos**

PSL Research University, Université Paris-Dauphine, LAMSADE, CNRS UMR 7243
{escoffier,eun-jung.kim,paschos}@lamsade.dauphine.fr

Abstract. Fixed-parameter algorithms, approximation algorithms and moderately exponential algorithms are three major approaches to algorithms design. While each of them being very active in its own, there is an increasing attention to the connection between different approaches. In particular, whether Independent Set would be better approximable once endowed with subexponential-time or fpt-time is a central question. In this paper, we present a strong link between the linear PCP conjecture and the inapproximability, thus partially answering this question.

1 Introduction

In this paper we look into three approaches to algorithms design: Fixed-parameter algorithms, approximation algorithms and moderately exponential algorithms. These three areas, each of them being very active in its own, have been considered as foreign to each other until recently. Polynomial-time approximation algorithm produces a solution whose quality is guaranteed to lie within a certain range from the optimum. One illustrative problem indicating the development of this area is Independent Set. The approximability of Independent Set within constant ratios[1] has remained as the most important open problems for a long time in the field. It was only after the novel characterization of the NP given by the PCP theorem [1, 2] that impossibility of such approximability has been definitely proven. Subsequent improvements of the original PCP theorem, leading to corresponding refinements of the characterization of NP have also led to the actual very strong inapproximability result for Independent Set, namely, that it is inapproximable within ratios $\Omega(n^{\varepsilon-1})$ for any $\varepsilon > 0$, unless P = NP [29].

Moderately exponential algorithm is to allow exponential running time for the sake of optimality. In this case, the endeavor lies in limiting the growth of running time function as slow as possible. Parameterized complexity provides

* Research supported by the French Agency for Research under the DEFIS program TODO, ANR-09-EMER-010

** Also, Institut Universitaire de France

[1] The approximation ratio of an algorithm computing a feasible solution for some problem is the ratio of the value of the solution computed over the optimal value for the problem.

an alternative framework to analyze the running time in a more refined way [14, 17]. The aim is to get an $O(f(k) \cdot n^{o(k)})$-time algorithm in which the exponent of the input size n is independent of k. As these two research programs offer a generous running time compared to polynomial-time approximation algorithms, a growing amount of attention is paid to them as a way to cope with hardness in approximability. The first one deals with *moderately exponential approximation*. The goal of this program is to explore approximability of highly inapproximable (in polynomial time) problems in superpolynomial or moderately exponential time. Roughly speaking, if a given problem is solvable in time say $O^*(\gamma^n)$ but it is NP-hard to approximate within some ratio r, we seek r-approximation algorithms with complexity - significantly - lower than $O^*(\gamma^n)$. This issue has been considered for several problems such as Set Cover [12, 6], Coloring [3, 5], Independent Set and Vertex Cover [7], Bandwidth [13, 18].

The second research program handles approximation by fixed parameter algorithms. In this approximation framework, we say that a parameterized (with parameter k) problem Π is r-approximable if there exists an algorithm taking as inputs an instance I of Π and k and either computes a solution smaller or greater than (depending on whether Π is, respectively, a minimization, or a maximization problem) rk, or returns "no", asserting in this case that there is no solution of value at most or at least k. This line of research was initiated by three independent works [15, 9, 11]. As an excellent overview in this direction, see [25].

Several natural questions can be asked dealing with these two programs. In particular, the following ones have been asked several times (see for instance [25, 15, 18, 7]) and of great interest:

Q1 can a highly inapproximable in polynomial time problem be well-approximated in subexponential time?

Q2 does a highly inapproximable in polynomial time problem become well-approximable in parameterized time?

Few answers have been obtained until now. Regarding **Q1**, negative results can be directly obtained by gap-reductions for certain problems. For instance, Coloring is not approximable within ratio $4/3 - \epsilon$, since this would allow to determine whether a graph is 3-colorable or not in subexponential time. This contradicts a widely-acknowledge computational assumption [22]:

> *Exponential Time Hypothesis* (ETH): There exists an $\epsilon > 0$ such that no algorithm solves 3Sat in time $2^{\epsilon n}$, where n is the number of variables.

Regarding **Q2**, [15] shows that assuming FPT$\neq$ W[2], for any r the Independent Dominating Set problem is not r-approximable (in FPT time)[2].

Among interesting problems for which **Q1** and **Q2** are worth being asked are Independent Set, Coloring and Dominating Set. They fit in the frame of both **Q1** and **Q2** above: they are hard to approximate in polynomial time while

[2] Actually, the result is even stronger: it is impossible to obtain a ratio $r = g(k)$ for any function g.

their approximability in subexponential or in parameterized time is still open. Note that Independent Set and Dominating Set are moderately exponential approximable within any ratio $1-\varepsilon$, for any $\varepsilon > 0$ [6, 7], while Coloring is approximable within ratio $(1 + 1/\chi(G))$, where $\chi(G))$ denotes the chromatic number of a graph G in moderately exponential time [3, 5].

Our contribution in this paper is to establish a link between a major conjecture in PCP theorem and inapproximability in subexponential-time and in fpt-time, assuming ETH. We first state the conjecture while the definition of PCP is deferred to the next section.

> *Linear PCP Conjecture* (LPC): 3Sat $\in \mathrm{PCP}_{1,1/2}[\log n + D, E]$, where n is the number of variables in the 3Sat instance, D and E are constant.

Unlike ETH which is arguably recognized as a valid statement, LPC is a wide open question. In the main results summarized below, we claim that if LPC turns out to hold, it immediately implies that one of the most interesting questions in subexponential and parameterized approximation is negatively answered.

Theorem 1. (Main Result) *Assuming* ETH*, the followings hold for* Independent Set *on n vertices, for any constant $0 < r < 1$.*

(i) There is no r-approximation algorithm in time $O(2^{n^{1-\delta}})$ for any $\delta > 0$.
(ii) There is no r-approximation algorithm in time $O(2^{o(n)})$ if LPC *holds.*
(iii) There is no r-approximation algorithm in time $O(f(k)n^{O(1)})$ if LPC *holds.*

Note that (i) is not conditional upon LPC. In fact, this is an immediate consequence of near-linear PCP construction achieved recently in [26]. It has led to inapproximability results for Max-3Sat and Max-3Lin for some subexponential running time under ETH. Section 2 reviews the known consequences of near-linear PCP. In Section 3, we present new results along this line. In Section 4, we consider parameterized inapproximability of two fundamental problems Independent Set and Dominating Set, provided LPC holds.

2 Preliminaries

2.1 PCP and inapproximability of Max-3Sat

A problem is in $\mathrm{PCP}_{\alpha,\beta}[q, p]$ if there exists a PCP verifier which uses q random bits, reads at most p bits in the proof and is such that:

- if the instance is positive, then there exists a proof such that V(erifier) accepts with probability at least α;
- if the instance is negative, then for any proof V accepts with probability at most β.

Based upon the above definition, the following theorem is proved in [26], presenting a further refinement of the characterization of NP.

Theorem 2. *[26] **(2-query PCP)** For every $\epsilon > 0$,*

$$3\text{SAT} \in \text{PCP}_{1,\epsilon}[(1 + o(1)) \log n + O(\log(1/\epsilon)), poly(1/\epsilon)]$$

Theorem 2 has some important corollaries in polynomial approximation. Among those, the following two are of particular interest in what follows.

Corollary 1. *[26] Under* ETH*, for every $\epsilon > 0$, and $\delta > 0$, it is impossible to distinguish between instances of* MAX 3-LIN *with m equations where at least $(1-\epsilon)m$ are satisfiable from instances where at most $(1/2+\epsilon)m$ are satisfiable, in time $O(2^{m^{1-\delta}})$.*

Corollary 2. *[26] Under* ETH*, for every $\epsilon > 0$, and $\delta > 0$, it is impossible to distinguish between instances of* MAX-3SAT *with m clauses where at least $(1-\epsilon)m$ are satisfiable from instances where at most $(7/8+\epsilon)m$ are satisfiable, in time $O(2^{m^{1-\delta}})$.*

The following is a stronger version of Corollary 2: it holds if LPC holds. This will be our working hypothesis.

Hypothesis 1 *Under* ETH*, there exists $r < 1$ such that: for every $\epsilon > 0$ it is impossible to distinguish between instances of* MAX-3SAT *with m clauses where at least $(1-\epsilon)m$ are satisfiable from instances where at most $(r+\epsilon)m$ are satisfiable, in time $2^{o(m)}$.*

A very standard argument gives the following fact.

Lemma 1. *If LPC holds, then Hypothesis 1 also hold.*

Proof. Suppose that $3\text{SAT} \in \text{PCP}_{1,1/2}[\log n + D, E]$, where n is the number of variables in 3SAT instance, D and E are constants.

Then the size of the proof is at most $E2^{|R|} = cn$ for some constant c (where $|R| = \log n + D$ is the number of random bits) since $E2^{|R|}$ is the total number of bits that we read in the proof. Take one variable for each bit in the proof: $x_1, \cdots, x_{cn}$. For each random string R: take all the 2^E possibilities for the E variables read, and write a CNF formula which is satisfied if and only if the verifier accepts. This can be done with a formula with a constant number of clauses, say C_1, each clause having a constant number of variables, say C_2 (C_1 and C_2 depends on E).

If we consider the CNF formed by all theses CNF for all the random clauses, we get a CNF with $C_1 2^{|R|}$ clauses on variables $x_1, \cdots, x_{cn}$. The clauses are on C_2 variables but by adding a constant number of variables we can replace a clause on C_2 variables by an equivalent set of clauses on 3 variables. This way we get a 3-CNF formula and multiply the number of variables and the number of clauses by a constant, so they are still linear in n. For each R you have a set of say C_1' clauses.

Suppose that we start from a satisfiable instance of 3SAT. Then there exists a proof for which the verifier always accepts. By taking the corresponding values

for the variables x_i, and extending it properly to the new variables y, all the clauses are satisfied.

Suppose that we start from a non satisfiable instance of 3SAT. Then for any proof (i.e. any truth values of variables), the verifier rejects for at least half of the random strings. If the verifier rejects for a random string R, then in the set of clauses corresponding to this variable at least one clause is not satisfied. It means that among the $C_1' 2^{|R|}$ clauses (total number of clauses), at least $1/2 \cdot 2^{|R|}$ are not satisfied, ie a fraction $1/(2C_1')$ of the clauses.

Then either $m = C_1' 2^{|R|} = O(n)$ clauses are satisfiable, or at least $m/(2C_1')$ clauses are not satisfied by each assignment. Distinguishing between these sets in time $2^{o(m)}$ would solve 3SAT in $2^{o(n)}$. □

Dealing with INDEPENDENT SET, it is easy to see that, for any increasing and unbounded function $r(n)$, the problem is approximable within ratio $1/r(n)$ in subexponential time (recall that ratios $n^{\epsilon-1}$ are are very unlikely to be achieved in polynomial time). Indeed, simply consider all the subsets of V of size at most $n/r(n)$ and return the largest independent set among these sets. If a maximum independent set has size at most $n/r(n)$ then the algorithm finds it, otherwise the algorithm outputs a solution of size $n/r(n)$, while the size of an optimum solution is at most n. The running time of the algorithm is $O^*(\binom{n}{n/r(n)})$ that is subexponential in n.

Let us finally note that INDEPENDENT SET has the so called self-improvement property [20] claiming, roughly speaking, that either it is polynomially approximable by a polynomial time approximation schema, or no polynomial algorithm exists that guarantees some constant approximation ratio, unless P = NP.

With a similar proof, the above self-improvement property can be proved for INDEPENDENT SET also in the case of parameterized approximation.

Lemma 2. *[16] The following statements are equivalent for* INDEPENDENT SET*:*

- *there exists* $r \in (0,1)$ *such that there exists an* r*-approximation parameterized algorithm;*
- *for any* $r \in (0,1)$ *there exists an* r*-approximation parameterized algorithm.*

2.2 Expander Graphs

Expander graphs is a well known tool that has been used to amplify the inapproximability gap for INDEPENDENT SET in polynomial time. Similar arguments allow actually to derive a gap amplification with linear size amplification of the instance (see Theorem 4 below). This will be useful later to derive inapproximability bounds in subexponential time (where linear size amplification plays a crucial role).

Definition 1. *A graph* G *is a* (n, d, α)*-expander graph if (i)* G *has* n *vertices, (ii)* G *is* d*-regular, (iii) all the eigenvalues* λ *of* G *but the largest one is such that* $|\lambda| \leqslant \alpha d$*.*

Fact 1. *For any $k \in \mathbb{N}^*$ and any $\alpha > 0$ there exists d and a (k^2, d, α)-expander graph. Moreover, d depends only on α, and this graph can be computed in polynomial time for every fixed α.*

This fact follows from the following lemmas.

Lemma 3 ([19], or Th. 8.1 in [21]). *For every positive integer k, there exists a $(k^2, 8, 5\sqrt{2}/8)$-expander graph, computable in polynomial time.*

If G is a graph with adjacency matrix M, let us denote G^k the graph with adjacency matrix M^k.

Lemma 4 (Fact 1.2 in [27]). *If G is a (n, d, α)-expander graph, then G^k is a (n, d^k, α^k)-expander graph.*

Proof. G^k is obviously d^k regular, and the eigenvalues of G^k are the eigenvalues of G to the power of k. □

Proof of Fact 1. Fix some $\alpha > 0$ and let p be the smallest integer such that $(5\sqrt{2}/8)^p \leqslant \alpha$. Then, G^p is as required. □

Let G be a graph on n vertices and H be a (n, d, α)-expander graph. Let t be a positive integer. We build the graph G'_t on $N = nd^{t-1}$ vertices: each vertex corresponds to a $(t-1)$-random walk $x = (x_1, \cdots, x_t)$ on H (meaning that x_1 is chosen at random, and x_{i+1} is chosen randomly in the set of neighbors of x_i), and two vertices $x = (x_1, \cdots, x_t)$ and $y = (y_1, \cdots, y_t)$ in G'_t are adjacent iff $\{x_1, \cdots, x_t, y_1, \cdots, y_t\}$ is a clique in G.

Theorem 3 (claims 3.15 and 3.16 in [21]). *Let G be a graph on n vertices and H be a (n, d, α)-expander graph. If $b > 6\alpha$, then:*

- *If $\omega(G) \leqslant bn$ then $\omega(G'_t) \leqslant (b + 2\alpha)^t N$;*
- *If $\omega(G) \geqslant bn$ then $\omega(G'_t) \geqslant (b - 2\alpha)^t N$.*

We are now able to prove the gap amplification with linear size amplification.

Theorem 4. *Let G be a graph on n vertices (for a sufficiently large n) and $a > b$ be two positive real numbers. Then for any real $r > 0$ one can build in polynomial time a graph G_r such that:*

- *G_r has $N \leqslant Cn$ vertices for C independent of G (C may depend on r);*
- *If $\omega(G) \leqslant bn$ then $\omega(G_r) \leqslant b_r N$;*
- *If $\omega(G) \geqslant an$ then $\omega(G_r) \geqslant a_r N$;*
- *$b_r/a_r \leqslant r$.*

Proof. Let $k = \lceil \sqrt{n} \rceil$. We modify G by adding $k^2 - n$ dummy (isolated) vertices. Let G' be the new graph. It has $n' = k^2$ vertices. Note that $n' \leqslant (\sqrt{n} + 1)^2 = n + 2\sqrt{n} + 1 = n + o(n)$. Let n be such that $1 - \epsilon \leqslant n/n' \leqslant 1$ for a small ϵ. Thanks to Fact 1, we consider a (k^2, d, α)-expander graph H for a sufficiently small α (the value of which will be fixed later). According to Theorem 3 (applied on G') we build in polynomial time a graph G'_t on $N = n'd^t$ vertices such that (choosing $\alpha < b/6$):

– If $\omega(G) \leqslant bn$ then $\omega(G') = \omega(G) \leqslant bn'$, hence $\omega(G'_t) \leqslant (b+2\alpha)^t N$;
– If $\omega(G) \geqslant an$ then $\omega(G') = \omega(G) \leqslant an'(1-\epsilon)$, hence $\omega(G'_t) \geqslant (a(1-\epsilon) - 2\alpha)^t N$.

We choose ϵ and α such that $a(1-\epsilon) - 2\alpha > b + 2\alpha$, and then t such that $(a(1-\epsilon)-2\alpha)^t/(b+2\alpha)^t \leqslant r$. The number of vertices of G'_t is clearly linear in n (first point of the theorem). Furthermore, $b_r = (b+2\alpha)^t$ and $a_r = (a(1-\epsilon)-2\alpha)^t$ fulfil items 2, 3 and 4. □

3 On the approximability of Independent Set and related problems in subexponential time

As mentioned in Section 2, an almost-linear size PCP construction [26] for 3Sat allows to get the negative results stated in Corollaries 1 and 2. In this section, we present further consequences of Theorem 2, based upon a combination of known reductions with (almost) linear size amplifications of the instance.

First, Theorem 2 combined with the reduction in [1] showing inapproximability results for Independent Set in polynomial time, leads to the following result.

Theorem 5. *Under* ETH*, for any $r > 0$ and any $\delta > 0$, there is no r-approximation algorithm for* Independent Set *running in time $O(2^{N^{1-\delta}})$, where N is the size of the input graph for* Independent Set.

Proof. Consider $r > 0$ and $\delta > 0$. Given an instance ϕ of 3 sat on n variables, we use the fact that $3\text{Sat} \in \text{PCP}_{1,r}[(1+o(1))\log n + D_r, E_r]$ (where D_r and E_r are constants that depends only on r) to build the following graph G_ϕ (see also [1]):

– for any random string R, and any possible values of the E_r bits read by V, add a vertex in the graph if V accepts;
– if two vertices are such that they have at least one contradicting bit (they read the same bit which is 1 for one of them and 0 for the other one), add an edge between them.

In particular, the set of vertices corresponding to the same random string is a clique.

Assume that ϕ is satisfiable. Then there exists a proof for which the verifier accepts for any random string R. Take for each random string R the vertex in G_ϕ corresponding to this proof. There is no conflict (no edge) between any of these $2^{|R|}$ vertices, hence $\alpha(G_\phi) = 2^{|R|}$ (where, in a graph G, $\alpha(G)$ denotes the size of a maximum independent set).

If ϕ is not satisfiable, then $\alpha(G_\phi) \leqslant r2^{|R|}$. Indeed, suppose that there is an independent set of size $\alpha > r2^{|R|}$. This independent set corresponds to a set of bits with no conflict, defining part of a proof that we can arbitrarily extend to a proof $\mathbb{P}$. The independent set has α vertices corresponding to α random strings (for which V accepts), meaning that the probability of acceptance for

this proof $\mathbb{P}$ is at least $\alpha/2^{|R|} > r$, a contradiction with the property of the verifier.

Furthermore, G_ϕ has $N \leqslant 2^{|R|}2^{E_r} \leqslant Cn^{1+o(1)}$ vertices (for some constant C). Then, one can see that, for any $r' > r$, an r'-approximation algorithm for Independent Set running in time $O(2^{N^{1-\delta}})$ would solve 3 sat in time $O(2^{n^{1-\delta'}})$ for some $\delta' < \delta$, contradicting ETH. □

Since (for $k \leqslant N$), $N^{k^{1-\delta}} = O(2^{N^{1-\delta'}})$, for some $\delta' < \delta$, the following result also holds.

Corollary 3. *Under* ETH*, for any $r > 0$ and any $\delta > 0$, there is no r-approximation algorithm for* Independent Set *(parameterized by k) running in time $O(N^{k^{1-\delta}})$, where N is the size of the input graph.*

The results of Theorem 5 and Corollary 3 can be immediately extended to problems that are linked to Independent Set by approximability preserving reductions (that preserve at least constant ratios) and have linear amplifications of the sizes of the instances.

For instance, this is the case of Set Packing (preservation of constant ratios and of ratios functions of the input size with amplification that is the identity function). This holds for the Max Bipartite Subgraph problem where, given a graph $G(V, E)$, the goal is to find a maximum-size subset $V' \subseteq V$ such that the graph $G[V']$ is a bipartite graph. Consider the following reduction from Independent Set to Max Bipartite Subgraph ([28]). Let $G(V, E)$ be an instance of Independent Set of order n. Construct a graph $G'(V', E')$ for Max Bipartite Subgraph by taking two distinct copies of G (denote them by G_1 and G_2, respectively) and adding the following edges: a vertex v_{i_1} of copy G_1 is linked with a vertex v_{j_2} of G_2, if and only if either $i = j$ or $(v_i, v_j) \in E$. G' has $2n$ vertices. Let now S be an independent set of G. Then, obviously, taking the two copies of S in G_1 and G_2 induces a bipartite graph of size $2|S|$. Conversely, consider an induced bipartite graph in G' of size t, and take the largest among the two color classes. By construction it corresponds to an independent set in G, whose size is at least $t/2$ (note that it cannot contain two copies of the same vertex). So, any r-approximate solution for Max Bipartite Subgraph in G' can be transformed into an r-approximate solution for Independent Set in G. Observe finally that the size of G' is two times the size of G.

Proposition 1. *Under* ETH*, for any $r > 0$ and any $\delta > 0$, there is no r-approximation algorithm for either* Set Packing *or* Max Bipartite Subgraph *running in time $O(2^{n^{1-\delta}})$ in a graph of order n.*

Dealing with minimization problems, Theorem 5 and Corollary 3 can be extended to Coloring, thanks to the reduction given in [23]. Given a graph G whose vertex set is partitioned into K cliques each of size S, and given a prime number $q > S$, a graph H_q having the following properties can be built in polynomial time:

- the vertex set of H_q is partitioned into q^2K cliques, each of size q^3;

– $\alpha(H_q) \leqslant \max\{q^2\alpha(G); q^2(\alpha(G)-1)+K; qK\}$;
– if $\alpha(G) = K$ then $\chi(H_q) = q^3$.

Note that this reduction uses the particular structure of graphs produced in the inapproximability result in [1] (as in Theorem 5). Then, we deduce the following result.

Proposition 2. *Under* ETH*, for any $r > 1$ and any $\delta > 0$, there is no r-approximation algorithm for* Coloring *running in time $O(2^{n^{1-\delta}})$ in a graph of order n.*

Proof. Fix a ratio $r > 1$, and let $r_{IS} > 0$ be such that $r_{IS} + r_{IS}^2 \leqslant 1/r$. Start from the graph G_ϕ produced in the proof of Theorem 5 for ratio r_{IS}. The vertex set of G_ϕ is partitioned into $K = 2^{|R|}$ cliques, each of size at most 2^{E_r}. By adding dummy vertices (a linear number, since E_r is a fixed constant), we can assume that each clique has the same size $S = 2^{E_r}$, so the number of vertices in G_ϕ is $N = KS = 2^{|R|}2^{E_r}$.

Let $q > \max\{S, 1/r_{IS}\}$ be a prime number, and consider the graph H_q produced from G_ϕ by the reduction in [23] mentioned above. If ϕ is satisfiable, $\alpha(G_\phi) = K$ and then by the third property of the graph H_q, $\chi(H_q) = q^3$. Otherwise, by the second property $\alpha(H_q) \leqslant \max\{q^2\alpha(G_\phi); q^2(\alpha(G_\phi)-1)+K; qK\}$. Formula ϕ being not satisfiable, $\alpha(G_\phi) \leqslant r_{IS}K$. By the choice of q, $qK \leqslant q^2 r_{IS}K$, so $\alpha(H_q) \leqslant q^2 r_{IS}K + K = (q^2 r_{IS} + 1)K$. Since the number of vertices in H_q is Kq^5, we get that $\chi(H_q) \geqslant q^5/(q^2 r_{IS} + 1)$. The gap created for the chromatic number in the two cases is then at least:

$$\frac{q^5}{(q^2 r_{IS} + 1)q^3} = \frac{1}{r_{IS} + 1/q^2} \geqslant \frac{1}{r_{IS} + r_{IS}^2} \geqslant r$$

The result follows since H_q has Kq^5 vertices and q is a constant (that depends only on the ratio r and on the constant number of bits p read by V), so the size of H_q is linear in the size of G_ϕ. □

We consider the approximability of Vertex Cover and Min-Sat in subexponential time. The following statement provides a lower bound to such a possibility.

Proposition 3. *Under* ETH*, for any $r > 0$ and any $\delta > 0$, there is no $(7/6 - \epsilon)$-approximation algorithm for* Vertex Cover *running in time $O(2^{N^{1-\delta}})$ in graphs of order N.*

Proof. We combine Corollary 1 with the following classical reduction. Consider an instance I of max 3-lin on m equations. Build the following graph G_I:

– for any equation and any of the eight possible values of the 3 variables in it, add a vertex in the graph if the equation is satisfied;
– if two vertices are such that they have one contradicting variable (the same variable has value 1 for one vertex and 0 for the other one), then add an edge between them.

In particular, the set of vertices corresponding to the same equation is a clique. Note that each equation is satisfied by exactly 4 values of the variables in it. Then, the number of vertices in the graph is $N = 4m$. Consider an independent set S in the graph G_I. Since there is no conflict, it corresponds to a partial assignment that can be arbitrarily completed into an assignment τ for the whole system. Each vertex in S corresponds to an equation satisfied by τ (and S has at most one vertex per equation), so τ satisfies (at least) $|S|$ equations. Reciprocally, if an assignment τ satisfies α clauses, there is obviously an independent set of size α in G_I. Hence, if $(1-\epsilon)m$ equations are satisfiable, there exists an independent set of size at least $(1-\epsilon)m$, i.e., a vertex cover of size at most $N-(1-\epsilon)m = N(3/4+\epsilon/4)$. If at most $(1/2+\epsilon)m$ equations are satisfiable, then each vertex cover has size at least $N-(1/2+\epsilon)m = N(7/8-\epsilon/4)$. □

The result of Proposition 3 can be extended to the Min-Sat problem via the following reduction [24]. Given a graph G, build the following instance on Min-Sat. For each edge (v_i, v_j) add a variable x_{ij}. For each vertex v_i add a clause c_i. Variablen x_{ij} appears positively in c_i and negatively in c_j. Then, take a vertex cover V^* of size k; for any x_{ij} fix the variable to true if $v_i \in V^*$, to false otherwise. Consider a clause c_j with $v_j \notin V^*$. If $\overline{x_{ij}}$ is in c_j then v_i is in V^* hence x_{ij} is true; if x_{ji} is in c_j then, by construction, x_{ji} is false. So c_j is not satisfied, and the assignment satisfies at mostn k clauses. Conversely, consider a truth assignment that satisfies k clauses $c_{i_1}, \cdots, c_{i_k}$. Consider the vertex set $V^* = \{v_{i_1}, \cdots, v_{i_k}\}$. For an edge (v_i, v_j), if x_{ij} is set to true then c_i is satisfied and v_i is in V^*, otherwise c_j is satisfied and v_j is in V^*, so V^* is a vertex cover of size k. Since the number of clauses in the reduction equals the number of vertices in the initial graph, we get the following result.

Proposition 4. *Under* ETH*, for any $r > 0$ and any $\delta > 0$, there is no $(7/6-\epsilon)$-approximation algorithm for* Min-Sat *running in time $2^{m^{1-\delta}}$ in CNF formulæ with m clauses.*

All the results given in this section are valid under ETH and rule out some ratio in subexponential time of the form $2^{n^{1-\delta}}$. It is worth noticing that if LPC holds, then all these result would hold for *any* subexponential time.

Corollary 4. *If* LPC*, under* ETH *the negative results of Theorems 5, Propositions 1, 2 and 3 (resp. 4) hold for any time complexity $2^{o(n)}$ (resp. $2^{o(m)}$).*

Proof. Using LPC, the same proof as in Theorem 5 creates a graph on $N = O(n)$ variables with either an independent set of size αN (if ϕ is satisfiable) or a maximum independent set of size at most $\alpha/2N$ (if ϕ is not satisfiable). Then Theorem 4 allows to amplify this gap from $1/2$ to any constant $r > 0$ while preserving the linear size of the instance. Results for the other problems immediately follow from the same arguments as above. □

4 Parameterized inapproximability bounds

It is shown in [10] that, under ETH, for any function f no algorithm running in time $f(k)n^{o(k)}$ can determine whether there exists an independent set of size k,

or not, (in a graph with n vertices). A challenging question is to obtain a similar result for approximation algorithms for Independent Set. In the sequel, we propose a reduction from Max-3Sat to Independent Set that, based upon the negative result of Corollary 2, only gives a negative result for *some* function f (because Corollary 2 only avoids *some* subexponential running time). However, this reduction gives the desired inapproximability result if Hypothesis 1, which is an enforcement of Corollary 2, is used.

Based upon Hypothesis 1, the following parameterized inapproximability bound can be proved. Recall that Hypothesis 1 assumes ETH.

Theorem 6. *Under Hypothesis 1, for every $\epsilon > 0$, no parameterized approximation algorithm for* Independent Set *running in time $f(k)N^{o(k)}$ can achieve approximation ratio $r + \epsilon$ in graphs of order N.*

Proof. Suppose that such an algorithm exists for some $\epsilon > 0$. W.l.o.g., we can assume that f is increasing, and that $f(k) \geqslant 2^k$. Take an instance I of Max-3Sat, let K be an integer that will be fixed later, and do the following:

- Partition the m clauses into K groups $H_1, \cdots, H_K$ each of them containing, roughly, m/K clauses each.
- Each group H_i involves a number $s_i \leqslant 3m/K$ of variables. For all possible values of these variables, add a vertex in the graph G_I if these values satisfy at least $\lambda m/K$ clauses in H_i (the value of λ will also be fixed later).
- Add an edge between two vertices if they have one contradicting variable.

In particular the vertices corresponding to the same group of clauses form a clique. It is easy to see that the so-constructed graph contains $N \leqslant K2^{3m/K}$ vertices.

The following easy claim holds.

Claim. If a variable assignment satisfies at least $\lambda m/K$ clauses in at most s groups, then it satisfies at most $\lambda m + s(1-\lambda)m/K$ clauses.

Proof of claim. Consider an assignment as the one claimed in claim's statement. This assignment satisfies at most m/K clauses in at most s groups, and at most $\lambda m/K$ in the other $K - s$ groups, so in total at most $sm/K + (K - s)\lambda m/K = \lambda m + s(1-\lambda)m/K$, that completes the proof of the lemma. □ ◇

Now, let us go back to the proof of the theorem.

- Assume an independent set of size at least t in G_I. Then one can achieve a partial solution that satisfies at least $\lambda m/K$ clauses in at least t groups. So, at least $t\lambda m/K$ clauses are satisfiable. In other words, if at most $(r + \epsilon')m$ clauses are satisfiable, then a maximum independent set in G_I has size at most $K\frac{r+\epsilon'}{\lambda}$.
- Suppose that at least $(1-\epsilon')m$ clauses are satisfiable. Then, using Lemma 4, there exists a solution satisfying at least $\lambda m/K$ clauses in at least $\frac{1-\epsilon'-\lambda}{1-\lambda}K$ groups; otherwise, it should be $\lambda m + s(1-\lambda)m/K < (1-\epsilon')m$. Then, there exists an independent set of size $\frac{1-\epsilon'-\lambda}{1-\lambda}K$ in G_I.

Now, set $K = \lceil \phi(m)/(1-\epsilon^2) \rceil$ where ϕ is the inverse function of f (i.e., $\phi = f^{-1}$). Set also $\lambda = 1-\epsilon$, and $\epsilon' = \epsilon^3$. Run the assumed $(r+\epsilon)$-approximation parameterized algorithm for INDEPENDENT SET in G_I with parameter $k = (1-\epsilon^2)K$. Then:

- if at least $(1-\epsilon')m$ equations are satisfiable, there exists an independent set of size at least $\frac{1-\epsilon'-\lambda}{1-\lambda}K = (1-\epsilon^3/\epsilon)K = (1-\epsilon^2)K = k$; so, the algorithm must output an independent set of size at least $(r+\epsilon)k$;
- if at most $(r+\epsilon')$ equations are satisfiable, the size of an independent set is at most $K\frac{r+\epsilon'}{\lambda} = K\frac{r+\epsilon^3}{1-\epsilon} = k\frac{r+\epsilon^3}{(1-\epsilon)(1-\epsilon^2)} = k(r+r\epsilon+o(\epsilon))$.

So, for ϵ sufficiently small, the algorithm allows to distinguish between the two cases of MAX-3SAT (for ϵ').

The running time of the yielded algorithm is $f(k)N^{o(k)}$, but $f(k) = f((1-\epsilon^2)K) = m$, and $N^{o(k)} = N^{k/\psi(k)}$ for some increasing and unbounded function ψ, and $N^{o(k)} = (K2^{3m/K})^{k/\psi(k)} = 2^{o(m)}$. □

Using Lemma 2 together with Theorem 6, the following result can be easily derived.

Corollary 5. *Under Hypothesis 1, for any $r \in (0,1)$ there is no r-approximation parameterized algorithm (i.e., an algorithm that runs in time $f(k)p(n)$ for some function f and some polynomial p).*

Let us now deal with DOMINATING SET that is known to be W[2]-hard [14]. Existence of fpt-approximation algorithms for this problem is an open question [15]. Here, we present an approximation preserving reduction (fitting the parameterized framework) that works with the special set of instances produced in the proof of Theorem 6. This reduction will allow us to obtain a lower bound (based on the same hypothesis) for the approximation of MIN DOMINATING SET from Theorem 6.

Consider a graph $G(V,E)$ on n vertices where V is a set of K cliques $C_1, \cdots, C_K$. We build a graph $G'(V',E')$ such that G has an independent set of size α if and only if G' has a dominating set of size $2K-\alpha$. The graph G' is built as follows:

- For each clique C_i in G, add a clique C'_i of the same size in G'. Add also:
  - an independent set S_i of size $3K$, each vertex in S_i being adjacent to all vertices in C'_i;
  - a special vertex t_i adjacent to all the vertices in C'_i.
- For each edge $e = (u,v)$ with u and v *not* in the same clique in G, add an independent set W_e of size $3K$. Suppose that $u \in C_i$ and $v \in C_j$. Then, each vertex in W_e is linked to t_i and to all vertices in C'_i but u (and t_j and all vertices in C'_j but v).

Informally, the reduction works as follows. The set S_i ensures that we have to take at least one vertex in each C'_i, the fact that $|W_e| = 3K$ ensures that it is never interesting to take a vertex in W_e. If we take vertex t_i in a dominating set,

this will mean that we do not take any vertex in the set C_i in the corresponding independent set in G. If we take one vertex in C'_i (but not t_i), this vertex will be in the independent set in G. Let us state this property in the following lemma.

Lemma 5. *G has an independent set of size α if and only if G' has a dominating set of size $2K - \alpha$.*

Proof. Suppose that G has an independent set S of size α. Then, S has one vertex in α sets C_i, and no vertex in the other $K - \alpha$ sets. We build a dominating set T in G' as follows: for each vertex in S we take its copy in G'. For each clique C_i without vertices in S, we take t_i and one (anyone) vertex in C'_i. The dominating set T has size $\alpha + 2(K - \alpha) = 2K - \alpha$. For each C'_i there exists a vertex in T; so, vertices in C'_i, t_i and vertices in S_i are dominated. Now take a vertex in W_e with $e = (u, v)$, $u \in C_i$ and $v \in C_j$. If $C_i \cap S = \emptyset$ (or $C_j \cap S = \emptyset$), then $t_i \in T$ (or $t_j \in T$) and, by construction, t_i is adjacent to all vertices in W_e. Otherwise, there exist $w \in S \cap C_i$ and $x \in S \cap C_j$. Since S is an independent set, either $w \neq u$ or $x \neq v$. If $w \neq u$, by construction w (its copy in C'_i) is adjacent to all vertices in W_e and, similarly, for x if $x \neq v$. So, T is a dominating set.

Conversely, suppose that T is a dominating set of size $2K - \alpha$. Since S_i is an independent set of size $3K$, we can assume that $T \cap S_i = \emptyset$ and the same occurs with W_e. In particular, there exists at least one vertex in T in each C_i. Now, suppose that T has two different vertices u and v in the same C_i. Then we can replace v by t_i getting a dominating set (vertices in S_i are still dominated by u, and any vertex in some W_e which is adjacent to v is adjacent to t_i). So, we can assume that T has the following form: exactly one vertex in each C_i, and $K - \alpha$ vertices t_i. Hence, there are α C'_i cliques where t_i is not in T. We consider in G the set S constituted by the α vertices in T in these α sets. Take two vertices u and v in S with, say, $u \in C'_i$ and $v \in C'_j$ (with $t_i \notin T$ and $t_j \notin T$). If there were an edge $e = (u, v)$ in G, neither u nor v would have dominated a vertex in W_e (by construction). Since neither t_i nor t_j is in T, this set would not have been a dominating set, a contradiction. So S is an independent set. □

Theorem 7. *Under Hypothesis 1, for every $\epsilon > 0$, no approximation algorithm running in time $f(k)N^{o(k)}$ can achieve approximation ratio smaller than $2-r-\epsilon$ for* Dominating Set *in graphs of order N.*

Proof. In the proof of Theorem 6, we produce a graph G_I which is made of K cliques and such that: if at least $(1 - \epsilon)m$ clauses are satisfiable in I, then there exists an independent set of size $(1 - O(\epsilon))K$; otherwise (at most $(r + \epsilon)m$ clauses are satisfiable in I), the maximum independent set has size at most $(r + O(\epsilon))K$. The previous reduction transforms G_I in a graph G'_I such that, applying Lemma 5, in the first case there exists a dominating set of size at most $2K - (1 - O(\epsilon))K = K(1 + O(\epsilon))$ while, in the second case, the size of a dominating set is at least $2K - (r + O(\epsilon))K = K(2 - r - O(\epsilon))$. Thus, we get a gap with parameter $k' = K(1 + O(\epsilon))$. Note that the number of vertices in G'_I is $N' = N + K + 3K + 3K|E_I| = O(N^3)$ (where E_I is the set of edges in G_I). If we were able to distinguish between these two sets of instances in time

$f(k')N'^{o(k')}$, this would allow to distinguish the corresponding independent set instances in time $f(k')N'^{o(k')} = g(k)N^{o(k)}$ since $k' = K(1+O(\epsilon)) = k(1+O(\epsilon))$ ($k = K(1-\epsilon^3)$ being the parameter chosen for the graph G_I). □

Such a lower bound immediately transfers to Set Cover since a graph on n vertices for Dominating Set can be easily transformed into an equivalent instance of min set cover with ground set and set system both of size n.

Corollary 6. *Under Hypothesis 1, for every $\epsilon > 0$, no approximation algorithm running in time $f(k)m^{o(k)}$ can achieve approximation ratio smaller than $2-r-\epsilon$ for* min set cover *in instances with m sets.*

5 Conclusion

This paper presents conditional lower bounds of approximation ratio in subexponential and fpt-time. Assuming ETH, we prove inapproximability in time $2^{n^{1-\delta}}$ for any $\delta > 0$ for the problems such as: Independent Set, Set Packing, Max Bipartite Subgraph, Coloring, Vertex Cover. If Linear PCP Conjecture turns out to hold, even in time $2^{o(n)}$ we cannot approximate any better. Also assuming ETH, we proved that Linear PCP Conjecture implies that fpt-time inapproximabilty of Independent Set and Dominating Set. Here, let us note that, all the results about Independent Set immediately apply also to Clique.

Our effort in this paper is only a first step and we wish to motivate further research. There remains a range of problems to be tackled, among which we propose the followings.

- Our inapproximability results, in particular those in fpt-time, are conditional upon Linear PCP Conjecture. Is it possible to relax the condition to a more plausible one?
- Or, we dare ask whether (certain) inapproximability results in fpt-time imply strong improvement in PCP theorem. For example, would the converse of Lemma 1 hold?

Note finally that we have considered in this article constant approximation ratios. In this sense, Theorem 5 is "tight" with respect to approximation ratios since, as mentioned in Section 2, ratio $1/r(n)$ is achievable in subexponential time for any increasing and unbounded function r. However, dealing with parameterized approximation algorithms, achieving a non constant ratio is also an open question. More precisely, finding in fpt-time an independent set of size $g(k)$ when there exists an independent set of size k is not known for *any* unbounded and increasing function g.

References

1. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. In *Proc. FOCS'92*, pages 14–23, 1992.

2. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and intractability of approximation problems. *J. Assoc. Comput. Mach.*, 45(3):501–555, 1998.
3. A. Bjørklund, T. Husfeldt, and M. Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.
4. E. Bonnet, B. Escoffier, V. Th. Paschos, and E. Tourniaire. Parameterized algorithms for MAX $(k, n-k)$-CUT. Unpublished manuscript.
5. N. Bourgeois, B. Escoffier, and V. Th. Paschos. Efficient approximation of MIN COLORING by moderately exponential algorithms. *Inform. Process. Lett.*, 109(16):950–954, 2009.
6. N. Bourgeois, B. Escoffier, and V. Th. Paschos. Efficient approximation of MIN SET COVER by moderately exponential algorithms. *Theoret. Comput. Sci.*, 410(21-23):2184–2195, 2009.
7. N. Bourgeois, B. Escoffier, and V. Th. Paschos. Approximation of MAX INDEPENDENT SET, MIN VERTEX COVER and related problems by moderately exponential algorithms. *Discrete Appl. Math.*, 159(17):1954–1970, 2011.
8. L. Cai. Parameter complexity of cardinality constrained optimization problems. *The Computer Journal*, 51:102–121, 2008.
9. L. Cai and X. Huang. Fixed-parameter approximation: conceptual framework and approximability results. In H. L. Bodlaender and M. A. Langston, editors, *Proc. International Workshop on Parameterized and Exact Computation, IWPEC'06*, volume 4169 of *Lecture Notes in Computer Science*, pages 96–108. Springer-Verlag, 2006.
10. J. Chen, X. Huang, I. A. Kanj and G. Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.* 72(8): 1346-1367, 2006.
11. Yijia Chen, Martin Grohe, and Magdalena Grüber. On parameterized approximability. In *IWPEC*, pages 109–120, 2006.
12. M. Cygan, L. Kowalik, and M. Wykurz. Exponential-time approximation of weighted set cover. *Inform. Process. Lett.*, 109(16):957–961, 2009.
13. M. Cygan and M. Pilipczuk. Exact and approximate bandwidth. *Theoret. Comput. Sci.*, 411(40–42):3701–3713, 2010.
14. R. G. Downey and M. R. Fellows *Parameterized Complexity.* Monographs in Computer Science. Springer, 1999.
15. R. G. Downey, M. R. Fellows, and C. McCartin. Parameterized approximation problems. In H. L. Bodlaender and M. A. Langston, editors, *Proc. International Workshop on Parameterized and Exact Computation, IWPEC'06*, volume 4169 of *Lecture Notes in Computer Science*, pages 121–129. Springer-Verlag, 2006.
16. B. Escoffier, V. Th. Paschos, and E. Tourniaire. Moderately exponential and parameterized approximation: some structural results. Unpublished manuscript.
17. J. Flum and M. Grohe. *Parameterized complexity theorey.* Texts in Theoretical Computer Science. Springer, 2006.
18. M. Fürer, S. Gaspers, and S. P. Kasiviswanathan. An exponential time 2-approximation algorithm for bandwidth. In *Proc. International Workshop on Parameterized and Exact Computation, IWPEC'09*, volume 5917 of *Lecture Notes in Computer Science*, pages 173–184. Springer, 2009.
19. O. Gabber and Z. Galil. Explicit constructions of linear-sized superconcentrators. J. Comput. System Sci., 22(3):407ÂŰ420, 1981.
20. M. R. Garey and D. S. Johnson. *Computers and intractability. A guide to the theory of NP-completeness.* W. H. Freeman, San Francisco, 1979.
21. S. Hoory, N. Linial and A. Widgerson. Expander graphs and their applications. Bulletin of the AMS. 43 (4), 439–561, 2006.

22. R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. System Sci.*, 63(4):512–530, 2001.
23. C. Lund and M. Yannakakis. On the Hardness of Approximating Minimization Problems. *J. ACM*, 41(5): 960-981, 1994.
24. M. V. Marathe and S. S. Ravi. On Approximation algorithms for the minimum satisfiability problem. *Inf. Process. Lett.* 58(1): 23-29, 1996.
25. D. Marx. Parameterized complexity and approximation algorithms. *The Computer Journal*, 51(1):60–78, 2008.
26. D. Moshkovitz and R. Raz. Two query pcp with sub-constant error. In *Proc. FOCS'08*, pages 314–323, 2008.
27. O. Reingold, S. P. Vadhan and A. Wigderson. Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders and Extractors. Electronic Colloquium on Computational Complexity (ECCC) 8(18): (2001).
28. H. U. Simon. On approximate solutions for combinatorial optimization problems. *SIAM J. Disc. Math.*, 3(2):294–310, 1990.
29. D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proc. STOC'06*, pages 681–690, 2006.