On Subexponential and FPT-Time Inapproximability*

Edouard Bonnet, 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 these different frameworks. In particular, whether Independent Set would be better approximable once endowed with subexponential-time or FPT-time is a central question. In this article, we provide new insights to this question using two complementary approaches; the former makes a strong link between the linear PCP conjecture and inapproximability; the latter builds a class of equivalent problems under approximation in subexponential time.

1 Introduction

Fixed-parameter algorithms, approximation algorithms and moderately exponential/subexponential algorithms are major approaches for efficiently solving NP-hard problems. 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 has remained as the most important open problems for a long time in the field. It was only after the novel characterization of **NP** by PCP theorem [2] that such inapproximability was proven assuming $\mathbf{P} \neq \mathbf{NP}$. Subsequent improvements of the original PCP theorem led to much stronger result for Independent Set: it is inapproximable within ratios $\Omega(n^{\varepsilon-1})$ for any $\varepsilon > 0$, unless $\mathbf{P} = \mathbf{NP}$ [3].

Moderately exponential (subexponential, respectively) computation allows exponential (subexponential, respectively) running time for the sake of optimality. In this case, the endeavor lies in limiting the growth of the running time function as slow as possible. Parameterized complexity provides an alternative framework to analyze the running time in a more refined way [4]. Given an instance with

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

** Also, Institut Universitaire de France.

G. Gutin and S. Szeider (Eds.): IPEC 2013, LNCS 8246, pp. 54–65, 2013.

a parameter k, the aim is to get an $O(f(k) \cdot n^c)$-time (or equivalently, FPT-time) algorithm for some constant c, where the constant c is independent of k. As these two research programs offer a generous running time when compared to that of classic approximation algorithms, a growing amount of attention is paid to them as a way to cope with hardness in approximability. The first one yields *moderately exponential approximation.* In moderately exponential approximation, the core question is whether a problem is approximable in moderately exponential time while such approximation is impossible in polynomial time. Suppose a problem is solvable in time $O^*(\gamma^n)$, but it is NP-hard to approximate within ratio r. Then, we seek for r-approximation algorithms of running time significantly faster than $O^*(\gamma^n)$. This issue has been considered for several problems [5,6,7,13,17].

The second research program handles approximation by fixed parameter algorithms. We say that a minimization (maximization, respecitvely) problem Π, together with a parameter k, is *parameterized r-approximable* if there exists an FPT-time algorithm which computes a solution of size at most (at least, respectively) rk whenever the input instance has a solution of size at most (at least, respectively) k. This line of research was initiated by three independent works [15,9,11]. For an excellent overview, see [22]. In what follows, parameterization means "standard parameterization", i.e., where problems are parameterized by the cost of the optimal solution.

Several natural questions can be asked dealing with these two programs. In particular, the following ones have been asked several times [22,15,17,7].

Q1: can a problem, which is highly inapproximable in polynomial time, be well-approximated in subexponential time?
Q2: does a problem, which is highly inapproximable in polynomial time, become well-approximable in FPT-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-acknowledged computational assumption [19]:

> *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[1] in FPT-time.

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 their approximability in subexponential or in parameterized time is still open.

In this paper, we study parameterized and subexponential (in)approximability of natural optimization problems. In particular, we follow two guidelines:

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

(i) getting inapproximability results under some conjecture and (ii) establishing classes of uniformly inapproximable problems under approximability preserving reductions.

Following the first direction, we establish a link between a major conjecture in PCP theorem and inapproximability in subexponential-time and in FPT-time, assuming ETH. Just below, we state this conjecture while the definition of PCP is deferred to the next section.

> *Linear PCP Conjecture* (LPC): 3Sat $\in \text{PCP}_{1,\beta}[\log|\phi| + D, E]$ for some $\beta \in (0,1)$, where $|\phi|$ is the size of the 3Sat instance (sum of lengths of clauses), D and E are constant.

Unlike ETH which is arguably recognized as a valid statement, LPC is a wide open question. In Lemma 1 stated in Section 2, we claim that if LPC turns out to hold, it implies that one of the most interesting questions in subexponential and parameterized approximation is answered in the negative. In particular, the followings hold for Independent Set on n vertices, for *any* constant $0 < r < 1$ assuming ETH:

(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.

Remark that (i) is not conditional upon LPC. In fact, this is an immediate consequence of near-linear PCP construction achieved in [14]. Note that similar inapproximability results under ETH for Max-3Sat and Max-3Lin for some subexponential running time have been obtained in [24].

Following the second guideline, we show that a number of problems are equivalent with respect to approximability in subexponential time. Designing a family of equivalent problems is a common way to provide an evidence in favor of hardness of these problems. One prominent example is the family of problems complete under SERF-reducibility [19] which leads to equivalent formulations of ETH. More precisely, for a given problem Π, let us formulate the following hypothesis, which can be seen as the approximate counterpart of ETH.

Hypothesis 1 (APETH(Π)). *There exist two constants $\epsilon > 0$ and r ($r < 1$ if Π is a maximization problem, $r > 1$, otherwise), such that Π is not r-approximable in time $2^{\epsilon n}$.*

We prove that several well-known problems are equivalent with respect to the APETH (APETH-equivalent). To this end, a notion called the *approximation preserving sparsification* is proposed. A recipe to prove that two problems A and B are APETH-equivalent consists of two steps. The first is to reduce an instance of A into a family of instances in "bounded" version (bounded degree for graph problems, bounded occurrence for satisfiability problems), which are equivalent with respect to approximability. This step is where the proposed notion comes into play. The second is to use standard approximability preserving reductions to derive equivalences between bounded versions of A and B. In this paper, we consider L-reductions [25] for this purpose. Furthermore, we show that if

APETH fails for one of these problems, then *any* problem in MaxSNP would be approximable for *any* constant ratio in subexponential FPT-time $2^{o(k)}$, which is also an evidence toward the validity of APETH. This result can be viewed as an extension of [10], which states that none of MaxSNP hard problems allows $2^{o(k)}$-time algorithm under ETH.

Some preliminaries and notation are given in Section 2. Results derived from PCP and LPC are given in Section 3. The second direction on equivalences between problems is described in Section 4.

2 Preliminaries and Notation

We denote by $\mathrm{PCP}_{\alpha,\beta}[q,p]$ (see for instance [2] for more on PCP systems) the set of problems for which there exists a PCP verifier which uses q random bits, reads at most p bits in the proof and is such that: (1) if the instance is positive, then there exists a proof such that V(erifier) accepts with probability at least α; (2) if the instance is negative, then for any proof V accepts with probability at most β. The following theorem is proved in [14] (see also Theorem 7 in [24]), presenting a further refinement of the characterization of NP.

Theorem 1. *[14] For every $\epsilon > 0$,*

$$3\textsc{Sat} \in \mathrm{PCP}_{1,\epsilon}[(1+o(1))\log n + O(\log(1/\epsilon)), O(\log(1/\epsilon))]$$

A recent improvement [24] of Theorem 1 (a PCP Theorem with two-query projection tests, sub-constant error and almost-linear size) has some important corollaries in polynomial approximation. In particular:

Corollary 1. *[24] 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 $O(2^{m^{1-\delta}})$.*

Under LPC, a stronger version of this result follows from standard argument[2].

Lemma 1. *If* LPC[3] *and ETH hold, then 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 $2^{o(m)}$.*

This (conditional) hardness result of approximating Max-3Sat will be the basis of the negative results of parameterized approximation in Section 3.1.

Let us now present two useful gap amplification results for Independent Set. First, as noted in [16], the so-called self-improvement property [18] can be proven for Independent Set also in the case of parameterized approximation.

[2] All missing proofs can be found in the extended version of the paper [1].

[3] Note that LPC as expressed in this article implies the result even with replacing $(1-\epsilon)m$ by m. However, we stick with this lighter statement $(1-\epsilon)m$ in order, in particular, to emphasize the fact that perfect completeness is not required in the LPC conjecture.

Lemma 2. *[16] If there exists a parameterized r-approximation algorithm for some $r \in (0,1)$ for* Independent Set*, then this is true for any $r \in (0,1)$.*

It is also well known that the very powerful tool of expander graphs allows to derive the following gap amplification for Independent Set (see [1]).

Theorem 2. *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 and specify constants a_r and b_r such that: (i) G_r has $N \leqslant Cn$ vertices, where C is some constant independent of G (but may depend on r); (ii) if $\omega(G) \leqslant bn$ then $\omega(G_r) \leqslant b_r N$; (iii) if $\omega(G) \geqslant an$ then $\omega(G_r) \geqslant a_r N$; (iv) $b_r/a_r \leqslant r$.*

Finally, we will use in the sequel the well known sparsification lemma [19]. Intuitively, this lemma allows to work with 3-SAT formula with linear lengths (the sum of the lengths of clauses is linearly bounded in the number of variables).

Lemma 3. *[19] For all $\epsilon > 0$, a 3-SAT formula ϕ on n variables can be written as the disjunction of at most $2^{\epsilon n}$ 3-SAT formula ϕ_i on (at most) n variables such that ϕ_i contains each variable in at most c_ϵ clauses for some constant c_ϵ. Moreover, this reduction takes at most $p(n)2^{\epsilon n}$ time.*

3 Some Consequences of (Almost-)Linear Size PCP System

3.1 Parameterized Inapproximability Bounds

It is shown in [12] 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 1, only gives a negative result for *some* function f (because Corollary 1 only avoids *some* subexponential running times). However, this reduction gives the inapproximability result sought, if the consequence of LPC given in Lemma 1 (which strengthens Corollary 1 and seems to be a much weaker assumption than LPC) is used instead. We emphasize the fact that the results in this section are valid as soon as a hardness result for Max-3Sat as that in Lemma 1 holds.

The proof of the following theorem essentially combines the parameterized reduction in [12] and a classic gap-preserving reduction.

Theorem 3. *Under* LPC *and* ETH*, there exists $r < 1$ such that, no parameterized approximation algorithm for* Independent Set *running in time $f(k)n^{o(k)}$ can achieve approximation ratio r in graphs of order n.*

The following result follows from Lemma 2 and Theorem 3.

Corollary 2. *Under* L*PC and* ETH*, for any* $r \in (0, 1)$ *there is no* r*-approximation parameterized algorithm for* Independent Set *(i.e., an algorithm that runs in time* $f(k)p(n)$ *for some function* f *and some polynomial* p*).*

Let us now consider Dominating Set which is known to be W[2]-hard [4]. The existence of parameterized approximation algorithms for this problem is open [15]. Here, we present an approximation preserving reduction (fitting the parameterized framework) which, given a graph $G(V, E)$ on n vertices where V is a set of K cliques $C_1, \cdots, C_K$, builds 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 $2K - \alpha$. Using the fact that the graphs produced in the proof of Theorem 3 are of this form (vertex set partitioned into cliques), this reduction will allow us to obtain a lower bound (based on the same hypothesis) for the approximation of min dominating set from Theorem 3.

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 and 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 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 4. *G has an independent set of size α if and only if G' has a dominating set of size $2K - \alpha$.*

Theorem 4. *Under* L*PC and* ETH*, there exists an* $r > 1$ *such that there is no* r*-approximation algorithm for* Dominating Set *running in time* $f(k)n^{o(k)}$ *where* n *is the order of the graph.*

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 Set Cover with ground set and set system both of size n.

Corollary 3. *Under* L*PC and* ETH*, there exists* $r > 1$ *such that there is no* r*-approximation algorithm for* Set Cover *running in time* $f(k)m^{o(k)}$ *in instances with* m *sets.*

3.2 On the Approximability of Independent Set and Related Problems in Subexponential Time

As mentioned in Section 2, an almost-linear size PCP construction [24] for 3Sat allows to get the negative result stated in Corollary 1. In this section, we present further consequences of Theorem 1, based upon a combination of known reductions with (almost) linear size amplifications of the instance.

First, Theorem 1 combined with the reduction in [2] showing inapproximability results for INDEPENDENT SET in polynomial time and the gap amplification of Theorem 2, 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 order of the input graph.*

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

Corollary 4. *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 order of the input graph.*

The results of Theorem 5 and Corollary 4 can be immediately extended to problems that are linked to INDEPENDENT SET by approximability preserving reductions (that preserve at least constant ratios) that have linear amplifications of the sizes of the instances, as in the following proposition.

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

Dealing with minimization problems, Theorem 5 and Corollary 4 can be extended to COLORING, using the reduction given in [21]. Note that this reduction uses the particular structure of graphs produced in the inapproximability result in [2] (as in Theorem 5). Hence, the following result can be derived.

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.*

Concerning the approximability of VERTEX COVER and MIN-SAT in subexponential time, the following holds.

Proposition 3. *Under* ETH*, for any $\epsilon > 0$ 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, nor 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 ratios in subexponential time of the form $2^{n^{1-\delta}}$. It is worth noticing that if LPC holds, then all these results would hold for *any* subexponential time. Note that this is in some sense optimal since it is easy to see that, for any increasing and unbounded function $r(n)$, INDEPENDENT SET is approximable within ratio $1/r(n)$ in subexponential time (simply consider all the subsets of V of size at most $n/r(n)$ and return the largest independent set among these sets).

Corollary 5. *If* L*PC holds, under* ETH *the negative results of Theorem 5 and Propositions 1, 2 and 3 hold for any time complexity $2^{o(n)}$.*

4 Subexponential Approximation Preserving Reducibility

In this section, we study subexponential approximation preserving reducibility. Recall that APETH(Π) (Hypothesis 1) states that it is hard to approximate in subexponential time problem Π, within some constant ratio r. We exhibit that a set of problems are APETH-equivalent using the notion of *approximation preserving sparsification*. We then link APETH with approximation in subexponential FPT-time.

4.1 Approximation Preserving Sparsification and APETH Equivalences

Recall that the sparsification lemma for 3SAT reduces a formula ϕ to a set of formulae ϕ_i with bounded occurrences of variables such that solving the instances ϕ_i would allow to solve ϕ. We attempt to build an analogous construction for subexponential approximation using the notion of *approximation preserving sparsification*. Given an optimization problem Π and some parameter of the instance, Π-B denotes the problem restricted to instances where the parameter is at most B. For example, we can prescribe the maximum degree of a graph or the maximum number of literal occurrences as the parameter. Then Π-B would be the problems restricted to instances with the parameter bounded by B.

Definition 1. *An approximation preserving sparsification from a problem Π to a bounded parameter version Π-B of Π is a pair (f, g) of functions such that, given any $\epsilon > 0$ and any instance I of Π:*

1. *f maps I into a set $f(I, \epsilon) = (I_1, I_2, \ldots, I_t)$ of instances of Π, where $t \leqslant 2^{\epsilon n}$ and $n_i = |I_i| \leqslant n$; moreover, there exists a constant B_ϵ (independent on I) such that any I_i has parameter at most B_ϵ;*
2. *for any $i \leqslant t$, g maps a solution S_i of an instance I_i (in $f(I, \epsilon)$) into a solution S of I;*
3. *there exists an index $i \leqslant t$ such that if a solution S_i is an r-approximation in I_i, then $S = g(I, \epsilon, I_i, S_i)$ is an r-approximation in I;*
4. *f is computable in time $O^*(2^{\epsilon n})$, and g is polynomial with respect to $|I|$.*

With a slight abuse of notation, let APETH(Π-B) denote the hypothesis: $\exists B$ such that APETH(Π-B), meaning that Π is hard to approximate in subexponential time even for some bounded parameter family of instances. Then the following holds[4].

Theorem 6. *If there exists an approximation preserving sparsification from Π to Π-B, then* APETH(Π) *if and only if* APETH(Π-B).

[4] Note that we could consider a more general definition, leading to the same theorem, by allowing (1) a slight amplification of the size of I_i ($n_i \leqslant \alpha n$ for some fixed α in item 1), (2) an expansion of the ratio in item 3 (if S_i is r-approximate S is $h(r)$ approximate where $h(r)$ goes to one when r goes to one) and (3) a computation time $O^*(2^{\epsilon n})$ for g in item 4.

We now illustrate this technique on some problems. It is worth noticing that the sparsification lemma for 3Sat in [19] is *not* approximation preserving[5]; one cannot use it to argue that approximating Max-3Sat (in subexponential time) is equivalent to approximating Max-3Sat with bounded occurrences.

Proposition 4. *There exists an approximation preserving sparsification from* Independent Set *to* Independent Set-B *and one from* Vertex Cover *to* Vertex Cover-B.

Proof. Let $\epsilon > 0$. It is well known that the positive root of $1 = x^{-1} + x^{-1-B}$ goes to one when B goes to infinity. Then, consider a B_ϵ such that this root is at most 2^ϵ. Our sparsification is obtained via a branching tree: the leaves of this tree will be the set of instances I_i; f consists of building this tree; a solution of an instance in the leaf corresponds, via the branching path leading to this leaf, to a solution of the root instance, and that is what g makes.

More precisely, for Independent Set, consider the following usual branching tree, starting from the initial graph G: as long as the maximum degree is at least B_ϵ, consider a vertex v of degree at least B_ϵ, and branch on it: either take v in the independent set (and remove $N[v]$), or do not take it. The branching stops when the maximum degree of the graph induced by the unfixed vertices is at most $B_\epsilon - 1$. When branching, at least $B_\epsilon + 1$ vertices are removed when taking v, and one when not taking v; thus the number of leaves is $t \leqslant 2^{\epsilon n}$ (by the choice of B_ϵ). Then, f and g satisfy items 1 and 2 of the definition. For item 3, it is sufficient to note that g maps S_i in S by adding adequate vertices. Then, if we consider the path in the tree corresponding to an optimal solution S^*, leading to a particular leaf G_i, we have that $|S^*| = |S^* \cap G_i| + k$ for some $k \geqslant 0$, and the solution S computed by g is of size $|S| = |S_i| + k$. So, $\frac{|S|}{|S^*|} \geqslant \frac{|S_i|}{|S^* \cap G_i|} \geqslant r$ if S_i is an r-approximation for G_i. The same argument holds also for Vertex Cover. □

Analogous arguments apply more generally to any problem where we have a "sufficiently good" branching rule when the parameter is large. Indeed, suppose we can ensure the decrease in instance size by $g(B)$ for nondecreasing and unbounded function g in all (possibly except for one) branches. Then such a branching rule can be utilized to yield an approximation preserving sparsification as in Proposition 4.

We give another approximation preserving sparsification, where there is no direct branching rule allowing to remove a sufficiently large number of vertices. Let Generalized Dominating Set be defined as follows: given a graph $G = (V, E)$ where V is partitioned into V_1, V_2, V_3, we ask for a minimum size set of vertices $V' \subseteq V_1 \cup V_2$ which dominates all vertices in $V_2 \cup V_3$. Of course, the case $V_2 = V$ corresponds to the usual Dominating Set problem. Note that Generalized Dominating Set is also a generalization of Set Cover, with $V_2 = \emptyset$, V_3 being the ground set and V_1 being the set system.

[5] One of the reasons is that when a clause C is contained in a clause C', a reduction rule removes C', that is safe for the satisfiability of the formula, but not when considering approximation.

Proposition 5. *There exists an approximation preserving sparsification from* Generalized Dominating Set *to* Generalized Dominating Set-B*.*

Combining Proposition 5 with some reductions, the following can be shown.

Lemma 5. APETH*(*Dominating Set*) implies* APETH*(*Independent Set-B*).*

Note that similarly, APETH(Set Cover) implies APETH(Independent Set-B), when the complexity of Set Cover is measured by $n + m$.

Then, we have the following set of equivalent problems.

Theorem 7. Set Cover*,* Independent Set*,* Independent Set-B*,* Vertex Cover*,* Vertex Cover-B*,* Dominating Set*,* Dominating Set-B*,* Max Cut-B*,* 3Sat-B*,* Max-kSat-B *(for any $k \geqslant 2$) are* APETH*-equivalent.*

Proof. The equivalences between Vertex Cover-B, Independent Set-B, Max Cut-B, 3Sat-B, Max-2Sat-B, Dominating Set-B follow immediately from [25]. Indeed, for these problems [25] provides L-reductions with linear size amplification. The equivalence between Max-kSat-B problems is also well known (just replace a clause of size k by $k - 1$ clauses of size 3).

The equivalence between Independent Set and Independent Set-B, Vertex Cover and Vertex Cover-B follows from Proposition 4. Finally, Lemma 5 allows us to conclude for Dominating Set. □

4.2 APETH and Parameterized Approximation

The equivalence drawn in Theorem 7 gives a first intuition that the corresponding problems should be hard to approximate in subexponential time for some ratio. In this section we show another argument towards this hypothesis: if it fails, then *any* MaxSNP problem admits for *any* $r < 1$ a parameterized r-approximation algorithm in subexponential time $2^{o(k)}$, which would be quite surprising. The following theorem can be construed as an extension of [10].

Theorem 8. *The following statements are equivalent:*

(i) APETH(Π) holds for one (equivalently all) problem(s) in Theorem 7;
(ii) there exist a MaxSNP-complete problem Π, some ratio $r < 1$ and a constant $\epsilon > 0$ such that there is no parameterized r-approximation algorithm for Π with running time $O(2^{\epsilon k} poly(|I|))$;
(iii) for any MaxSNP-complete problem Π, there exist a ratio $r < 1$ and an $\epsilon > 0$ such that there is no parameterized r-approximation algorithm for Π with running time $O(2^{\epsilon k} poly(|I|))$.

As an interesting complement of the above theorem, we show that trade-offs between (exponential) running time and approximation ratio do exist for any MaxSNP problem. In [8], it is shown that every MaxSNP problem Π is fixed-parameter tractable in time $2^{O(k)}$ for the standard parameterization, while in [25] it is shown that Π is approximable in polynomial time within a constant ratio

ρ_Π. We prove here that there exists a family of parameterized approximation algorithms achieving ratio $\rho_\Pi + \epsilon$, for any $\epsilon > 0$, and running in time $2^{O(\epsilon k)}$. This is obtained as a consequence of a result in [20].

Proposition 6. *Let Π be a standard parameterization of a MaxSNP-complete problem. For any $\epsilon > 0$, there exists a parameterized $(\rho_\Pi + \epsilon)$-approximation algorithm for Π running in time $\gamma^{\epsilon k} \cdot poly(|I|)$ for some constant γ.*

5 Conclusion

More interesting questions remain untouched in the junction of approximation and (sub)exponential-time/FPT-time computations. This paper is only a first step in this direction and we wish to motivate further research. Among a range of problems to be tackled, we propose the followings.

- Our inapproximability results 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?
- Can we design approximation preserving sparsifications for problems like Max Cut or Max-3Sat? It seems to be difficult to design a sparsifier based on branching rules, so a novel idea is needed.

Note that we have considered in this article constant approximation ratios. As noted earlier, 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.

Finally, let us note that, in the same vein of the first part of our work, Mathieson [23] recently studied a proof checking view of parameterized complexity. Possible links between these two approaches are worth being investigated in future works.

References

1. Bonnet, É., Escoffier, B., Kim, E.J., Paschos, V.T.: On Subexponential and FPT-time Inapproximability. CoRR abs/1211.6656 (2012)
2. Arora, S., Lund, C., Motwani, R., Sudan, M., Szegedy, M.: Proof verification and intractability of approximation problems. J. ACM 45, 501–555 (1998)
3. Zuckerman, D.: Linear degree extractors and the inapproximability of max clique and chromatic number. In: Proc. STOC 2006, pp. 681–690 (2006)
4. Downey, R.G., Fellows, M.R.: Parameterized complexity. Monographs in Computer Science. Springer, New York (1999)

5. Björklund, A., Husfeldt, T., Koivisto, M.: Set partitioning via inclusion-exclusion. SIAM J. Comput. 39, 546–563 (2009)
6. Bourgeois, N., Escoffier, B., Paschos, V.T.: Efficient approximation of min coloring by moderately exponential algorithms. Inform. Process. Lett. 109, 950–954 (2009)
7. Bourgeois, N., Escoffier, B., Paschos, V.T.: Approximation of max independent set, min vertex cover and related problems by moderately exponential algorithms. Discrete Appl. Math. 159, 1954–1970 (2011)
8. Cai, L., Chen, J.: On fixed-parameter tractability and approximability of NP optimization problems. J. Comput. System Sci. 54, 465–474 (1997)
9. Cai, L., Huang, X.: Fixed-parameter approximation: Conceptual framework and approximability results. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 96–108. Springer, Heidelberg (2006)
10. Cai, L., Juedes, D.W.: On the existence of subexponential parameterized algorithms. J. Comput. Syst. Sci. 67(4), 789–807 (2003)
11. Chen, Y.-J., Grohe, M., Grüber, M.: On parameterized approximability. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 109–120. Springer, Heidelberg (2006)
12. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. J. Comput. System Sci. 72, 1346–1367 (2006)
13. Cygan, M., Pilipczuk, M.: Exact and approximate bandwidth. Theoret. Comput. Sci. 411, 3701–3713 (2010)
14. Dinur, I.: The PCP theorem by gap amplification. J. ACM 54 (2007)
15. Downey, R.G., Fellows, M.R., McCartin, C.: Parameterized approximation problems. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 121–129. Springer, Heidelberg (2006)
16. Escoffier, B., Paschos, V.T., Tourniaire, E.: Moderately exponential and parameterized approximation: some structural results (unpublished manuscript)
17. Fürer, M., Gaspers, S., Kasiviswanathan, S.P.: An exponential time 2-approximation algorithm for bandwidth. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 173–184. Springer, Heidelberg (2009)
18. Garey, M.R., Johnson, D.S.: Computers and intractability. In: A Guide to the Theory of NP-completeness, W. H. Freeman, San Francisco (1979)
19. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? J. Comput. System Sci. 63, 512–530 (2001)
20. Kim, E.J., Williams, R.: Improved parameterized algorithms for above average constraint satisfaction. In: Marx, D., Rossmanith, P. (eds.) IPEC 2011. LNCS, vol. 7112, pp. 118–131. Springer, Heidelberg (2012)
21. Lund, C., Yannakakis, M.: On the hardness of approximating minimization problems. J. Assoc. Comput. Mach. 41, 960–981 (1994)
22. Marx, D.: Parameterized complexity and approximation algorithms. The Computer Journal 51, 60–78 (2008)
23. Mathieson, L.: A proof checking view of parameterized complexity. CoRR abs/1206.2436 (2012)
24. Moshkovitz, D., Raz, R.: Two query PCP with sub-constant error. In: Proc. FOCS 2008, pp. 314–323 (2008)
25. Papadimitriou, C.H., Yannakakis, M.: Optimization, approximation and complexity classes. J. Comput. System Sci. 43, 425–440 (1991)
26. Vazirani, V.: Approximation algorithms. Springer, Berlin (2001)