The exponential time hypothesis and the parameterized clique problem

Yijia Chen[1], Kord Eickmeyer[2*], and Jörg Flum[3]

[1] Department of Computer Science, Shanghai Jiaotong University
yijia.chen@cs.sjtu.edu.cn
[2] National Institute of Informatics, Tokyo
eickmeye@nii.ac.jp
[3] Mathematisches Institut, Albert-Ludwigs-Universität Freiburg
joerg.flum@math.uni-freiburg.de

Abstract. In parameterized complexity there are three natural definitions of fixed-parameter tractability called strongly uniform, weakly uniform and nonuniform fpt. Similarly, there are three notions of subexponential time, yielding three flavours of the exponential time hierarchy (ETH) stating that 3SAT is not solvable in subexponential time. It is known that ETH implies that p-CLIQUE is not fixed-parameter tractable if both are taken to be strongly uniform or both are taken to be uniform, and we extend this to the nonuniform case. We also show that even weakly uniform subexponential time is strictly contained in nonuniform subexponential time. Furthermore, we deduce from nonuniform ETH that no single exponent d allows for arbitrarily good fpt-approximations of clique.

1. Introduction

In parameterized complexity, FPT most commonly denotes the class of *strongly uniformly* fixed-parameter tractable problems, i.e., parameterized problems solvable in time $f(k) \cdot n^{O(1)}$ for some *computable* function f. Downey and Fellows also introduced the classes $\mathrm{FPT}_{\mathrm{uni}}$ and $\mathrm{FPT}_{\mathrm{nu}}$ of *uniformly* and *nonuniformly* fixed-parameter tractable problems, where one drops the condition that f be computable or allows for different algorithms for each k, respectively. The obvious inclusions between these classes can be shown to be strict [1]. We give detailed definitions in Section 2.

In classical complexity, the subexponential time classes

$$\mathrm{DTIME}\left(2^{o^{\mathrm{eff}}(n)}\right), \quad \mathrm{DTIME}\left(2^{o(n)}\right), \quad \text{and} \quad \bigcap_{\varepsilon>0}\mathrm{DTIME}\left(2^{\varepsilon \cdot n}\right), \qquad (1)$$

have been considered. In particular, there are the corresponding versions of the exponential time hypothesis, namely the *strongly uniform exponential time hypothesis* ETH, the *uniform exponential time hypothesis* $\mathrm{ETH}_{\mathrm{uni}}$, and the *nonuniform exponential time hypothesis* $\mathrm{ETH}_{\mathrm{nu}}$, which are the statements that 3SAT is

* This work was supported by a fellowship of the second author within the FIT-Programme of the German Academic Exchange Service (DAAD).

not in these respective classes, where n denotes the number of variables of the input formula. By results due to Impagliazzo et al. [2] we know that these three statements are equivalent to the ones obtained by replacing 3SAT by the clique problem CLIQUE; then, n denotes the number of vertices of the corresponding graph.

Furthermore, it is known [3] that ETH implies p-CLIQUE $\notin$ FPT, where p-CLIQUE denotes the parameterized clique problem. As pointed out, for example in [4], there is a correspondence between subexponential algorithms having a running time $2^{o(n)}$ and $\mathrm{FPT}_{\mathrm{uni}}$ similar to that between running time $2^{o^{\mathrm{eff}}(n)}$ and FPT. Therefore, it is not surprising that $\mathrm{ETH}_{\mathrm{uni}}$ implies p-CLIQUE $\notin \mathrm{FPT}_{\mathrm{uni}}$ (we include a proof in Section 4).

The first main result of this paper shows that $\mathrm{ETH}_{\mathrm{nu}}$ implies that p-CLIQUE $\notin \mathrm{FPT}_{\mathrm{nu}}$. So, putting the three results together, we see that:

(i) if ETH holds, then p-CLIQUE $\notin$ FPT;
(ii) if $\mathrm{ETH}_{\mathrm{uni}}$ holds, then p-CLIQUE $\notin \mathrm{FPT}_{\mathrm{uni}}$;
(iii) if $\mathrm{ETH}_{\mathrm{nu}}$ holds, then p-CLIQUE $\notin \mathrm{FPT}_{\mathrm{nu}}$.

One of the most important complexity classes of (apparently) intractable parameterized problems is the class W[1], the class of problems (strongly uniformly) fpt-reducible to p-CLIQUE. By replacing (strongly uniformly) fpt-reducible by uniformly fpt-reducible and nonuniformly fpt-reducible, we get the classes $\mathrm{W}[1]_{\mathrm{uni}}$ and $\mathrm{W}[1]_{\mathrm{nu}}$, respectively. So, the previous results can be seen as providing some evidence that $\mathrm{FPT} \neq \mathrm{W}[1]$, $\mathrm{FPT}_{\mathrm{uni}} \neq \mathrm{W}[1]_{\mathrm{uni}}$, and $\mathrm{FPT}_{\mathrm{nu}} \neq \mathrm{W}[1]_{\mathrm{nu}}$. Note that the strongest separation, $\mathrm{FPT}_{\mathrm{nu}} \neq \mathrm{W}[1]_{\mathrm{nu}}$, obtained in this paper under $\mathrm{ETH}_{\mathrm{nu}}$, plays a role in [5] (see the comment following Theorem 1.1 of that paper).

We get the implication (iii) by applying Levin's theorem on optimal inverters. In fact, assuming that p-CLIQUE $\in \mathrm{FPT}_{\mathrm{nu}}$, we use the family of algorithms witnessing this fact to prove that a suitable optimal inverter shows that p-CLIQUE is in "$\mathrm{FPT}_{\mathrm{uni}}$ for positive instances". Once we have this single algorithm for p-CLIQUE, we adapt the techniques we used in [6] to show (i), to get that the clique problem CLIQUE is in $\bigcap_{\varepsilon>0} \mathrm{DTIME}\left(2^{\varepsilon \cdot n}\right)$, that is, the failure of $\mathrm{ETH}_{\mathrm{nu}}$.

The basic idea of parameterized approximability is explained in [7] as follows: "Suppose we have a problem that is hard to approximate. Can we at least approximate it efficiently for instances for which the optimum is small? The classical theory of inapproximability does not seem to help answering this question, because usually the hardness proofs require fairly large solutions." In the papers [7] and [8] the framework of parameterized approximability was introduced. In particular, the concept of an fpt approximation algorithm with a given approximation ratio was coined and some (in)approximability results were proven.

Applying Levin's result again, we obtain our second main result, a nonapproximability result for p-CLIQUE under the assumption $\mathrm{ETH}_{\mathrm{nu}}$; in fact, we show that $\mathrm{ETH}_{\mathrm{nu}}$ implies that for every $d \in \mathbb{N}$ there is a $\rho > 1$ such that p-CLIQUE has no parameterized approximation algorithm with approximation ratio ρ and running time $f(k) \cdot n^d$ for some function $f : \mathbb{N} \to \mathbb{N}$.

The content of the different sections is the following. We recall some concepts and fix some notation in Section 2. In Section 3, from Levin's theorem we derive a result on the clique problem used to obtain both main results (in Section 4 and Section 5). Then, in Section 6, we show that the results relating the complexity of p-CLIQUE and the different variants of ETH are of a more general nature. Finally, in Section 7, we show that the second and third time class in (1) are distinct.

2. Some preliminaries

By $\mathbb{N}$ we denote the positive natural numbers. If a function $f : \mathbb{N} \to \mathbb{N}$ is nondecreasing and unbounded, then we denote by f^{-1} the function $f^{-1} : \mathbb{N} \to \mathbb{N}$ defined by

$$f^{-1}(n) := \begin{cases} \max\{i \in \mathbb{N} \mid f(i) \leq n\}, & \text{if } n \geq f(1) \\ 1, & \text{otherwise.} \end{cases}$$

Then, $f(f^{-1}(n)) \leq n$ for all $n \geq f(1)$ and the function f^{-1} is nondecreasing and unbounded.

Let $f, g : \mathbb{N} \to \mathbb{N}$ be functions. Then $f \in o^{\text{eff}}(g)$ (often written as $f(n) \in o^{\text{eff}}(g(n))$) if there is a computable function $h : \mathbb{N} \to \mathbb{N}$ such that for all $\ell \geq 1$ and $n \geq h(\ell)$, we have $f(n) \leq g(n)/\ell$.

We denote the alphabet $\{0, 1\}$ by Σ. The length of a string $x \in \Sigma^*$ is denoted by $|x|$. We identify problems with subsets Q of Σ^*. If $x \in Q$ we say that x is a *positive* instance of the problem Q. Clearly, as done mostly, we present concrete problems in a verbal, hence uncodified form. For example, we introduce the problem CLIQUE in the form:

CLIQUE
Instance: A graph G and $k \in \mathbb{N}$.
Problem: Does G have a clique of size k?

A graph G is given by its vertex set $V(G)$ and its edge set $E(G)$. By $|G|$ we denote the length of a string naturally encoding G. The cardinality or size of a set S is also denoted by $|S|$.

If $\mathbb{A}$ is an algorithm and $\mathbb{A}$ halts on input x, then we denote by $t_{\mathbb{A}}(x)$ the number of steps of $\mathbb{A}$ on input x; if $\mathbb{A}$ does not halt on x, then $t_{\mathbb{A}}(x) := \infty$.

We view *parameterized problems* as pairs (Q, κ) consisting of a classical problem $Q \subseteq \Sigma^*$ and a *parameterization* $\kappa : \Sigma^* \to \mathbb{N}$, which is required to be polynomial time computable. We will present parameterized problems in the form we do for p-CLIQUE:

p-CLIQUE
Instance: A graph G and $k \in \mathbb{N}$.
Parameter: k.
Problem: Does G have a clique of size k?

A parameterized problem (Q, κ) is *(strongly uniformly) fixed-parameter tractable* or, in FPT, if there is an algorithm $\mathbb{A}$ deciding Q, a natural number d, and a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for all $x \in \Sigma^*$,

$$t_{\mathbb{A}}(x) \leq f(\kappa(x)) \cdot |x|^d.$$

If in this definition we do not require the computability of f, then (Q, κ) is *uniformly fixed-parameter tractable* or, in $\mathrm{FPT}_{\mathrm{uni}}$. Finally, (Q, κ) is *nonuniformly fixed-parameter tractable* or, in $\mathrm{FPT}_{\mathrm{nu}}$, if there is a natural number d and a function $f : \mathbb{N} \to \mathbb{N}$ such that for every $k \in \mathbb{N}$ there is an algorithm $\mathbb{A}_k$ deciding the set

$$\{x \in Q \mid \kappa(x) = k\}$$

with $t_{\mathbb{A}_k}(x) \leq f(\kappa(x)) \cdot |x|^d$ for all $x \in \Sigma^*$.

In Section 6 we assume that the reader is familiar with the notion of (strongly uniform) fpt-reduction, with the classes of the W-hierarchy, and for $t, d \in \mathbb{N}$ with the weighted satisfiability problem p-WSAT$(\Gamma_{t,d})$ (e.g., see [9]). We write $(Q, \kappa) \leq^{\mathrm{fpt}} (Q', \kappa')$ if there is an fpt-reduction from (Q, κ) to (Q', κ').

3. An application of Levin's Optimal Inverter

Let $F : \Sigma^* \to \Sigma^*$ be a function. An algorithm $\mathbb{A}$ *inverts* F if for every x in the *range* of F the algorithm $\mathbb{A}$ computes some w with $F(w) = x$. For x not in the range of F the algorithm $\mathbb{A}$ can behave arbitrarily.

Theorem 1 (Levin's Optimal Inverter Theorem [10]). *For every algorithm $\mathbb{F}$ computing a function $F : \Sigma^* \to \Sigma^*$ there exists an* optimal inverter (with respect to $\mathbb{F}$)*, that is, an algorithm $\mathbb{O}$ such that:*

(O1) $\mathbb{O}$ inverts F and $t_{\mathbb{O}}(y) = \infty$ for every input y not in the range of F;

(O2) for every inverter $\mathbb{I}$ of F there is a constant $a_{\mathbb{I}} \in \mathbb{N}$ such that for every y in the range of F we have

$$t_{\mathbb{O}}(y) \leq a_{\mathbb{I}} \cdot \big(|y| + t_{\mathbb{I}}(y) + t_{\mathbb{F}}(\mathbb{I}(y))\big)^2.$$

The following hypothesis $(*)$ on CLIQUE will be useful to get the two applications of Levin's result we have in mind.

Definition 2. We say that CLIQUE satisfies $(*)$ if

for every $\ell \in \mathbb{N}$ there is an $e_\ell \in \mathbb{N}$ such that for every $k \in \mathbb{N}$ there is a constant $a_{\ell,k} \in \mathbb{N}$ and an algorithm $\mathbb{A}_{\ell,k}$ which on every graph G decides whether G contains a clique of size k in time

$$a_{\ell,k} \cdot |G|^{e_\ell + k/\ell}.$$

Clearly:

Lemma 3. *If p-CLIQUE $\in \mathrm{FPT}_{\mathrm{nu}}$, then CLIQUE satisfies $(*)$.*

Assume that Clique satisfies $(*)$. Using the family of algorithms of $(*)$ we can show that an optimal inverter of an appropriate function essentially decides Clique with a running time "subexponential in k for positive instances". More precisely, we get:

Lemma 4. *If* Clique *satisfies* $(*)$*, then there is an algorithm* $\mathbb{A}$ *deciding* Clique *and there is a function* $f : \mathbb{N} \to \mathbb{N}$ *such that*

$$t_{\mathbb{A}}(G, k) \leq f(k) \cdot |G|^{o(k)}$$

for every positive *instance* (G, k) *of* Clique*.*

Proof. We consider the function $F : \Sigma^* \to \Sigma^*$ defined by

$$F(x) := \begin{cases} (G, |S|), & \text{if } x = (G, S) \text{ where } S \text{ is a clique in the graph } G \\ \lambda, & \text{otherwise.} \end{cases}$$

Here, λ denotes the empty string. Clearly, there is a linear time algorithm $\mathbb{F}$ computing F. Let $\mathbb{O}$ be an optimal inverter of F (with respect to $\mathbb{F}$).

As we assume that Clique satisfies $(*)$, we have for every $\ell \in \mathbb{N}$ an $e_\ell \in \mathbb{N}$ such that for every $k \in \mathbb{N}$ there is an algorithm $\mathbb{A}_{\ell,k}$ which on every graph G decides whether G contains a clique of size k in time $c_{\ell,k} \cdot |G|^{e_\ell + k/\ell}$ where $c_{\ell,k} \in \mathbb{N}$.

Let $\ell, k_0 \in \mathbb{N}$. We consider the following algorithm $\mathbb{B}_{\ell,k_0}$ (thereby assuming that every graph comes with an ordering of its vertices):

```
$\mathbb{B}_{\ell,k_0}$    // $G = (V(G), E(G))$ a graph and $k \in \mathbb{N}$

1. if $k \neq k_0$ then simulate $\mathbb{O}$ on $(G, k)$ else
2.      simulate $\mathbb{A}_{\ell,k_0}$ on $(G, k)$
3.      if the simulation rejects then never halt
4.      $S \leftarrow V(G)$
5.           for $i = 1$ to $|V(G)|$ do
6.                $v \leftarrow$ the $i$th vertex of $V(G)$
7.                simulate $\mathbb{A}_{\ell,k_0}$ on $G[S \setminus \{v\}]$ [4]
8.                if the simulation accepts then $S \leftarrow S \setminus \{v\}$
9.           output $(G, S)$.
```

It is easy to see that $\mathbb{B}_{\ell,k_0}$ inverts f. Furthermore, for every graph G with a clique of size k

$$t_{\mathbb{B}_{\ell,k_0}}(G, k_0) \leq c_{\ell,k_0} \cdot \left|G\right|^{e_\ell + 1 + k_0/\ell}. \tag{2}$$

[4] For a subset M of the vertex set $V(G)$ of a graph G, as usual, $G[M]$ denotes the subgraph of G induced on M.

Hence, there is a function $a : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ such that for every graph G containing a clique of size k, we have

$$\begin{aligned} t_{\mathbb{O}}(G,k) &\leq a(\ell,k) \cdot \big(|(G,k)| + t_{\mathbb{B}_{\ell,k}}(G,k) + t_{\mathbb{F}}(\mathbb{B}_{\ell,k}(G,k))\big)^2 && \text{(by (Thm1, O2))} \\ &\leq a(\ell,k) \cdot c_{\ell,k} \cdot |G|^{O(e_\ell + 1 + k/\ell)} && \text{(by (2)).} \end{aligned}$$

Thus, for suitable functions $h : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ and $d : \mathbb{N} \to \mathbb{N}$, we have

$$t_{\mathbb{O}}(G,k) \leq h(\ell,k) \cdot \left|G\right|^{O(d(\ell)+k/\ell)}$$

for every positive instance (G,k) of p-Clique. We can assume that the function d is nondecreasing and unbounded. Hence, for every positive instance (G,k),

$$\begin{aligned} t_{\mathbb{O}}(G,k) &\leq h\left(d^{-1}\left(\left\lceil\sqrt{k}\right\rceil\right), k\right) \cdot \left|G\right|^{O(d(d^{-1}(\lceil\sqrt{k}\rceil)) + k/d^{-1}(\lceil\sqrt{k}\rceil))} \\ &= h\left(d^{-1}\left(\left\lceil\sqrt{k}\right\rceil\right), k\right) \cdot |G|^{o(k)}. \end{aligned}$$

Now, essentially by running the algorithm $\mathbb{O}$ and a brute force algorithm solving Clique in parallel, we get an algorithm $\mathbb{A}$ as claimed in the statement of the lemma. □

4. ETH$_{\text{nu}}$ and the complexity of p-Clique

In this section we show our first main result, namely:

Theorem 5. *If* ETH$_{\text{nu}}$ *holds, then* p-Clique $\notin$ FPT$_{\text{nu}}$.

To obtain this result we prove the following chain of implications

$$\text{(a)} \Rightarrow \text{(b)}_{\text{pos}} \Rightarrow \text{(c)}_{\text{pos}} \Rightarrow \text{(d)},$$

where

(a) p-Clique $\in$ FPT$_{\text{nu}}$;

(b)$_{\text{pos}}$ There is an algorithm $\mathbb{A}$ deciding Clique such that for all *positive* instances (G,k) of Clique we have

$$t_{\mathbb{A}}(G,k) \leq f(k) \cdot \left|G\right|^{o(k)}$$

for some function $f : \mathbb{N} \to \mathbb{N}$.

(c)$_{\text{pos}}$ There is an algorithm $\mathbb{A}$ deciding Clique such that for all *positive* instances (G,k) of Clique, where G has vertex set $V(G)$, we have

$$t_{\mathbb{A}}(G,k) \leq 2^{o(|V(G)|)}.$$

(d) ETH$_{\text{nu}}$ does not hold.

Note that $\neg$(d) $\Rightarrow \neg$(a) is the claim of Theorem 5.

The implication (a) $\Rightarrow$ (b)$_{\text{pos}}$ was shown in the previous section (Lemma 3 and Lemma 4). We turn to the implication (b)$_{\text{pos}}$ $\Rightarrow$ (c)$_{\text{pos}}$. Let (b) and (c) be the statements obtained from (b)$_{\text{pos}}$ and (c)$_{\text{pos}}$, respectively, by deleting the restriction to positive instances. Note that (c) is equivalent to the failure of ETH$_{\text{uni}}$. Furthermore, we let (b)$_{\text{eff}}$ be the statement (b) with the additional requirement that the function f is computable and let (c)$_{\text{eff}}$ be the statement obtained from (c) by replacing $2^{o(|V(G)|)}$ by $2^{o^{\text{eff}}(|V(G)|)}$. Again note that (c)$_{\text{eff}}$ is equivalent to the failure of ETH.

Lemma 6. *(1)* (b)$_{\text{eff}}$ *implies* (c)$_{\text{eff}}$*;*
(2) (b) *implies* (c)*;*
(3) (b)$_{\text{pos}}$ *implies* (c)$_{\text{pos}}$.

Part (1) of this lemma was shown as Theorem 27 in [6] (and previously in [3]). We argue similarly to get parts (2) and (3). In particular, we use the following lemma stated and proved in [6] as Lemma 28. Its proof uses the fact that a clique in a graph G can be viewed as an "amalgamation of local cliques" of subgraphs of G.

Lemma 7. *There is an algorithm $\mathbb{D}$ that assigns to every graph $G = (V, E)$ and $k, m \leq |V|$ in time polynomial in $|V| \cdot 2^m$ a graph $G' = (V', E')$ with $|V'| \leq |V|^2 \cdot 2^m$ such that*

$$G \text{ has a clique of size } k \iff G' \text{ has a clique of size } \lceil |V|/m \rceil. \tag{3}$$

Proof of Lemma 6 (3): The proof of Lemma 6 (2) is obtained by the obvious modification and is left to the reader.

Let the algorithm $\mathbb{D}$ be as in the previous lemma. Assuming (b)$_{\text{pos}}$ there is an algorithm $\mathbb{A}$ deciding Clique such that for all positive instances (G, k) of Clique we have $t_{\mathbb{A}}(G, k) \leq f(k) \cdot |G|^{o(k)}$ and hence,

$$t_{\mathbb{A}}(G, k) \leq f(k) \cdot \big|V(G)\big|^{o(k)} \tag{4}$$

for some function $f : \mathbb{N} \to \mathbb{N}$. We consider the following algorithm deciding Clique:

```
$\mathbb{B}$        // $G$ a graph and $k \in \mathbb{N}$

1.  Do in parallel for every $m \leq |V(G)|$ the following
2.          simulate $\mathbb{D}$ on $(G, k, m)$ and let $G'$ be its output
3.          simulate $\mathbb{A}$ on $(G', \lceil |V(G)|/m \rceil)$
4.          **if** $\mathbb{A}$ accepts for some $m$ **then** accept
5.                  **else** reject.
```

Let (G, k) be a positive instance of Clique and $n := |V(G)|$. Without loss of generality, we can assume that f is nondecreasing and unbounded. For

$$m := \max\left\{ \left\lceil \frac{n}{f^{-1}(n)} \right\rceil, \lceil \log n \rceil \right\}$$

we have $m \geq \log n$ and $m \in o(n)$ and, by (4),

$$t_{\mathbb{A}}(G', \lceil |V(G)|/m \rceil) \leq f(\lceil n/m \rceil) \cdot (n^2 \cdot 2^m)^{o(n/m)} = 2^{o(n)}.$$

Thus, the running time for Line 2 to Line 5 is bounded by $2^{o(n)}$. Therefore

$$t_{\mathbb{B}}(G, k) \leq O(n \cdot 2^{o(n)}) \leq 2^{o(n)}. \qquad \square$$

We already remarked that (c) is equivalent to the failure of $\mathrm{ETH}_{\mathrm{uni}}$. Thus, part (2) of the previous lemma yields:

Corollary 8. *If* $\mathrm{ETH}_{\mathrm{uni}}$*, then* p-CLIQUE $\notin \mathrm{FPT}_{\mathrm{uni}}$.

Proof of $(\mathrm{c})_{\mathrm{pos}} \Rightarrow (\mathrm{d})$: For every $\varepsilon > 0$ there is an n_0 such that for graphs with $|V(G)| > n_0$ the running time of the algorithm asserted by $(\mathrm{c})_{\mathrm{pos}}$ is bounded by $2^{\varepsilon |V(G)|}$ on positive instances. For graphs with at least n_0 vertices we let the algorithm run for at most this many steps and reject if it does not hold within this time bound. For smaller graphs we use brute force.

For later purposes we remark:

Corollary 9. *If* CLIQUE *satisfies* $(*)$*, then* $\mathrm{ETH}_{\mathrm{nu}}$ *does not hold.*

Proof. If CLIQUE satisfies $(*)$, then $(\mathrm{b})_{\mathrm{pos}}$ holds by Lemma 4. We have shown that $(\mathrm{b})_{\mathrm{pos}}$ implies (d), thus, $\mathrm{ETH}_{\mathrm{nu}}$ does not hold. $\square$

5. $\mathrm{ETH}_{\mathrm{nu}}$ and the parameterized approximability of p-Clique.

Let $\rho > 1$ be a real number. As in [7], we say that an algorithm $\mathbb{A}$ is an *fpt*$_{uni}$ *parameterized approximation algorithm for* p-CLIQUE *with approximation ratio* ρ if

(i) $t_{\mathbb{A}}(G, k) \leq f(k) \cdot |V(G)|^{O(1)}$ for all instances (G, k) of p-CLIQUE and some function $f : \mathbb{N} \to \mathbb{N}$;
(ii) for all positive instances (G, k) of p-CLIQUE the algorithm $\mathbb{A}$ outputs a clique of size at least k/ρ; otherwise, the output of $\mathbb{A}$ can be arbitrary.

If $d \in \mathbb{N}$ and we get $t_{\mathbb{A}}(G, k) \leq f(k) \cdot |V(G)|^d$ in (i), then we say that $\mathbb{A}$ is an *fpt*$_{uni}$ *parameterized approximation algorithm for* p-CLIQUE *with approximation ratio* ρ *and exponent* d.

Now we can state the main result of this section:

Theorem 10. *If* $\mathrm{ETH}_{\mathrm{nu}}$ *holds, then for every* $d \in \mathbb{N}$ *there is a* $\rho > 1$ *such that* p-CLIQUE *has no fpt*$_{uni}$ *parameterized approximation algorithm with approximation ratio* ρ *and exponent* d.

The key observation that, together with Corollary 9, will yield this theorem is contained in the following lemma.

Lemma 11. *Assume that p-*Clique *has an fpt$_{\mathrm{uni}}$ parameterized approximation algorithm with approximation ratio $\rho > 1$ and exponent $d \geq 2$. Then, for every rational number r with $0 < r \leq \frac{1}{\log \rho}$, there is an algorithm $\mathbb{B}$ deciding* Clique *such that for some function $g : \mathbb{N} \to \mathbb{N}$ and every instance (G, k) of* Clique

$$t_{\mathbb{B}}(G, k) \leq g(k) \cdot |V(G)|^{r+2+d\cdot\lceil k/r\rceil}.$$

Proof. The main idea is as follows: we assume the existence of an fpt$_{\mathrm{uni}}$ parameterized approximation algorithm $\mathbb{A}$ for p-Clique. Given an instance (G, k) of Clique we stretch it by passing to an equivalent "product instance" (G', k'). By applying $\mathbb{A}$ to (G', k') we can decide whether $(G, k) \in$ Clique.

For a graph $G = (V, E)$ we let $\omega(G)$ be the size of a maximum clique in G. Furthermore, for every $m \in \mathbb{N}$ with $m \geq 1$ we denote by G^m the graph $(V(G^m), E(G^m))$, where

$$\begin{aligned} V(G^m) &:= V^m = \big\{(v_1, \ldots, v_m) \bigm| v_1, \ldots, v_m \in V\big\} \\ E(G^m) &:= \Big\{\big\{(u_1, \ldots, u_m), (v_1, \ldots, v_m)\big\} \\ &\qquad\qquad \Bigm| \{u_1, \ldots, u_m, v_1, \ldots, v_m\} \text{ is a clique in } G \\ &\qquad\qquad\qquad \text{and } (u_1, \ldots, u_m) \neq (v_1, \ldots, v_m)\Big\}. \end{aligned}$$

One easily verifies that

$$\omega(G^m) = \omega(G)^m. \tag{5}$$

Now we let $\mathbb{A}$ be an fpt$_{\mathrm{uni}}$ parameterized approximation algorithm for p-Clique with approximation ratio $\rho > 1$ and exponent $d \geq 2$, say, with running time bounded by $f(k) \cdot |V(G)|^d$. Let r be a rational number with $0 < r \leq \frac{1}{\log \rho}$. Then, $\rho \leq \sqrt[r]{2}$ and for every $k \in \mathbb{N}$ with $k \geq 2$ we get

$$\left(\frac{k}{k-1}\right)^{\lceil k/r\rceil} > \rho. \tag{6}$$

We let $\mathbb{B}$ be the following algorithm:

```
B      // $G$ a graph and $k \in \mathbb{N}$

1. if $k = 1$ or $k < r$ then decide whether $G$ has a clique of size $k$ by
   brute force
2.     else simulate $\mathbb{A}$ on $(G^{\lceil k/r\rceil}, k^{\lceil k/r\rceil})$
3.         if $\mathbb{A}$ outputs a clique of $G^{\lceil k/r\rceil}$ of size $k^{\lceil k/r\rceil}/\rho$
4.             then accept else reject.
```

The algorithm $\mathbb{B}$ decides Clique: Clearly, the answer is correct if $k = 1$ or $k < r$. So assume that $k \geq 2$ and $k \geq r$. If G has no clique of size k, that is, $\omega(G) \leq k - 1$, then, by (5), $\omega(G^{\lceil k/r\rceil}) \leq (k-1)^{\lceil k/r\rceil}$. By (6),

$$\frac{k^{\lceil k/r\rceil}}{\rho} > (k-1)^{\lceil k/r\rceil};$$

thus, compare Line 3 and Line 4, the algorithm $\mathbb{B}$ rejects (G,k). If $\omega(G) \geq k$ and hence, $\omega(G^{\lceil k/r \rceil}) \geq k^{\lceil k/r \rceil}$, then the approximation algorithm $\mathbb{A}$ outputs a clique of $G^{\lceil k/r \rceil}$ of size $k^{\lceil k/r \rceil}/\rho$; thus $\mathbb{B}$ accepts (G,k).

Moreover, on every instance (G,k) with $G = (V,E)$ the running time of $\mathbb{B}$ is bounded by

$$|V|^{r+2} + |V|^{2\cdot\lceil k/r \rceil+2} + f\left(k^{\lceil k/r \rceil}\right) \cdot \left|V\left(G^{\lceil k/r \rceil}\right)\right|^{d} \leq g(k) \cdot |V|^{r+2+d\cdot\lceil k/r \rceil},$$

for a suitable $g : \mathbb{N} \to \mathbb{N}$. □

Setting $r := 1/\log\rho$ in the previous lemma, we get:

Corollary 12. *If there is an fpt$_{uni}$ parameterized approximation algorithm for p-*Clique *with approximation ratio $\rho \geq 1$ and exponent $d \geq 2$, then there exists $e \in \mathbb{N}$ and an algorithm $\mathbb{B}$ deciding* Clique *with $t_{\mathbb{B}}(G,k) \leq g(k) \cdot |V(G)|^{e+d\cdot k\cdot\log\rho}$*

Proof of Theorem 10: By contradiction, assume that for some $d \geq 2$ and all $\rho > 1$ the problem p-Clique has an fpt$_{\text{uni}}$ parameterized approximation algorithm with approximation ratio ρ and exponent d.

If $\ell \in \mathbb{N}$, then $d \cdot \ell \leq \frac{1}{\log\rho}$ for suitable $\rho > 1$. Thus, by Lemma 11, there is an algorithm $\mathbb{A}_\ell$ deciding Clique such that for some $e_\ell \in \mathbb{N}$ and some function $g : \mathbb{N} \to \mathbb{N}$ and every instance (G,k)

$$t_{\mathbb{A}_\ell}(G,k) \leq g(k) \cdot |V(G)|^{e_\ell + k/\ell}.$$

Fix $k \in \mathbb{N}$. Clearly, then there is an algorithm $\mathbb{A}_{\ell,k}$ which on every graph G decides whether G contains a clique of size k in time

$$O\left(|G|^{e_\ell + k/\ell}\right).$$

Thus, Clique satisfies $(*)$ (the property introduced in Definition 2). Therefore, ETH$_{\text{nu}}$ does not hold by Corollary 9. □

6. Some extensions and generalizations

Some results of Section 3 and of Section 4 can be stated more succinctly and in a more general form in the framework of parameterized complexity theory. We do this in this section, at the same time getting some open questions.

The class FPT$_{\text{nu}}$ is closed under fpt-reductions, that is,

$$\text{if } (Q,\kappa) \leq^{\text{fpt}} (Q',\kappa') \text{ and } (Q',\kappa') \in \text{FPT}_{\text{nu}}, \text{ then } (Q,\kappa) \in \text{FPT}_{\text{nu}}. \tag{7}$$

Thus, for every W[1]-complete problem (Q,κ) (complete under fpt-reductions), we have

$$(Q,\kappa) \in \text{FPT}_{\text{nu}} \iff \text{W[1]} \subseteq \text{FPT}_{\text{nu}}. \tag{8}$$

Denote by $\mathrm{FPT}^+_{\mathrm{uni}}$ the class of problems (Q, κ) such that there is an algorithm deciding Q and with running time $h(\kappa(x)) \cdot |x|^{O(1)}$ for $x \in Q$, that is, for positive instances x of Q. The class $\mathrm{FPT}^+_{\mathrm{uni}}$ is closed under fpt-reductions, too. So, again we have for every W[1]-complete problem (Q, κ),

$$(Q, \kappa) \in \mathrm{FPT}^+_{\mathrm{uni}} \iff \mathrm{W}[1] \subseteq \mathrm{FPT}^+_{\mathrm{uni}}. \tag{9}$$

Corollary 13. *For every* W[1]*-complete problem* (Q, κ)*,*

$$(Q, \kappa) \in \mathrm{FPT}_{\mathrm{nu}} \quad \textit{implies} \quad (Q, \kappa) \in \mathrm{FPT}^+_{\mathrm{uni}}.$$

Proof. By Lemma 3 and Lemma 4, we know that the implication holds for the W[1]-complete problem p-Clique. Now, the claim follows by (8) and (9). □

It is not clear whether the previous implication holds for all problems $(Q, \kappa) \in$ W[1] (and not only for the complete ones). Of course, it does if FPT = W[1]. The proof of Lemma 4 makes essential use of a self-reducibility property of p-Clique. For $t, d \in \mathbb{N}$ the weighted satisfiability problem p-WSat$(\Gamma_{t,d})$ has this self-reducibility property, too. So, along the lines of Lemma 4, one gets (we leave the details to the reader):

Lemma 14. *Let* $t, d \in \mathbb{N}$*. Then*

$$p\text{-WSat}(\Gamma_{t,d}) \in \mathrm{FPT}_{\mathrm{nu}} \quad \textit{implies} \quad p\text{-WSat}(\Gamma_{t,d}) \in \mathrm{FPT}^+_{\mathrm{uni}}.$$

And thus, we get the extension of Corollary 13 to all levels of the W-hierarchy:

Proposition 15. *Let* $t \in \mathbb{N}$*. For every* W$[t]$*-complete problem* (Q, κ)*,*

$$(Q, \kappa) \in \mathrm{FPT}_{\mathrm{nu}} \quad \textit{implies} \quad (Q, \kappa) \in \mathrm{FPT}^+_{\mathrm{uni}}.$$

After Theorem 5, we have considered two further properties of the clique problem there denoted by $(\mathrm{b})_{\mathrm{pos}}$ and $(\mathrm{c})_{\mathrm{pos}}$. One could also define these properties for arbitrary parameterized problems (even though, there are some subtle points as the terms $2^{o(|V(G)|)}$ and $2^{o(|G|)}$ may be distinct). More importantly, these properties are not closed under fpt-reductions. So somehow one has to check whether other implications of Section 4 survive problem by problem. In Appendix A we do this for the most prominent W[2]-complete problem, namely the parameterized dominating set problem p-DS.

7. An example

We believe that the three statements ETH, $\mathrm{ETH}_{\mathrm{uni}}$, and $\mathrm{ETH}_{\mathrm{nu}}$ are true and hence equivalent. Here we consider the "underlying" complexity classes (see (1)). Clearly,

$$\mathrm{DTIME}\left(2^{o^{\mathrm{eff}}(n)}\right) \quad \subseteq \quad \mathrm{DTIME}\left(2^{o(n)}\right) \quad \subseteq \quad \textstyle\bigcap_{\varepsilon>0} \mathrm{DTIME}\left(2^{\varepsilon \cdot n}\right) \tag{10}$$

To the best of our knowledge it is open whether the first inclusion is strict. Here we show the strictness of the second inclusion in (10). We remark that in [6, Proposition 5] we proved that the first class, that is, the effective version of the second one, coincides with an effective version of the third class.

For $m \in \mathbb{N}$ let 1^m be the string in Σ^* consisting of m ones. Recall that $\Sigma = \{0, 1\}$. For a Turing machine $\mathbb{M}$ we denote by $\mathrm{enc}(\mathbb{M})$ a string in Σ^* reasonably encoding the Turing machine $\mathbb{M}$. Furthermore, $|\mathbb{M}|$ denotes the length of $\mathrm{enc}(\mathbb{M})$, $|\mathbb{M}| = |\mathrm{enc}(\mathbb{M})|$.

Theorem 16. *The problem*

Exp-Halt	
Instance:	A Turing machine $\mathbb{M}$, $x \in \Sigma^*$, and 1^m with $m \in \mathbb{N}$.
Problem:	Does $\mathbb{M}$ accept x in time $2^{\lfloor m/(|\mathbb{M}| + |x|) \rfloor}$?

is in $\bigcap_{\varepsilon>0} \mathrm{DTIME}\left(2^{\varepsilon \cdot n}\right) \setminus \mathrm{DTIME}\left(2^{o(n)}\right)$.

A proof can be found in Appendix B.

References

1. Downey, R., Fellows, M.: Fixed-parameter tractability and completeness iii: some structural aspects of the w hierarchy. In Ambos-Spies, K., Homer, S., Schöning, U., eds.: Complexity theory, New York, NY, USA, Cambridge University Press (1993) 191–225
2. Impagliazzo, R., Paturi, R., Zane, F.: Which problems have strongly exponential complexity? Journal of Computer and System Sciences **63** (2001) 512–530
3. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Linear fpt reductions and computational lower bounds. In: Proc. of STOC'04. (2004) 212–221
4. Flum, J., Grohe, M.: Parametrized complexity and subexponential time (column: Computational complexity). Bulletin of the EATCS **84** (2004) 71–100
5. Grohe, M.: The complexity of homomorphism and constraint satisfaction problems seen from the other side. J. ACM **54**(1) (2007)
6. Chen, Y., Flum, J.: On miniaturized problems in parameterized complexity theory. Theoretical Computer Science **351**(3) (2006) 314–336
7. Chen, Y., Grohe, M., Grüber, M.: On parameterized approximability. In: 2nd International Workshop on Parameterized and Exact Computation. Number 4169 in LNCS, Springer-Verlag (2006) 109–120
8. Marx, D.: Parameterized complexity and approximation algorithms. The Computer Journal **51** (2008) 60–78
9. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer-Verlag, Berlin Heidelberg (2006)
10. Levin, L.: Universal sequential search problems. Problems of Information Transmission **9**(3) (1973) 265–266

A. Uniform Subexponential Solvability of p-DS and DS

We consider the parameterized dominating set problem p-DS, which is one of the most prominent W[2]-complete problems:

> p-DS
> *Instance:* A graph G and $k \in \mathbb{N}$.
> *Parameter:* k.
> *Problem:* Does G have a dominating set of size k?

We denote by DS the underlying classical problem. In [6, Theorem 29] we have shown:

If p-DS *can be decided in time* $f(k) \cdot |V(G)|^{o^{\mathrm{eff}}(k)}$ *for some computable* $f : \mathbb{N} \to \mathbb{N}$*, then* DS *can be decided in time* $2^{o^{\mathrm{eff}}(|V(G)|)}$.

The reader should compare this result with the following one in the spirit of this paper.

Theorem 17. *If there is an algorithm* $\mathbb{A}$ *deciding* p-DS *such that for all positive instances* (G, k) *of* p-DS *we have*

$$t_{\mathbb{A}}(G, k) \leq f(k) \cdot \left|G\right|^{o(k)}$$

for some function $f : \mathbb{N} \to \mathbb{N}$*, then there is an algorithm* $\mathbb{B}$ *deciding* DS *such that for all positive instances* (G, k) *of* DS *we have*

$$t_{\mathbb{B}}(G, k) \leq 2^{o(|V(G)|)}.$$

The proof of the corresponding result for Clique, namely the implication $(\mathrm{b})_{\mathrm{pos}} \Rightarrow (\mathrm{c})_{\mathrm{pos}}$, was based on Lemma 7 which used the fact that a clique in a graph can be viewed as an "amalgamation of local cliques" of subgraphs. As dominating sets are not necessarily an "amalgamation of local dominating sets," in [6] we took a detour via the weighted satisfiability problem for propositional formulas in CNF. As an inspection of the exposition in [6] shows, it can be adapted to a proof of Theorem 17.

B. Proof of Theorem 16

Proof. First, for $\ell \in \mathbb{N}$, we show that Exp-Halt $\in$ DTIME$\left(2^{n/\ell}\right)$. For this purpose let S_ℓ be the set of all triples $(\mathbb{M}, x, s)$, where $\mathbb{M}$ is a Turing machine, $x \in \Sigma^*$, and $s \in \mathbb{N}$ such that

$$|\mathbb{M}| + |x| \leq \ell \quad \text{and} \quad \mathbb{M} \text{ accepts } x \text{ in exactly } s \text{ steps.}$$

Clearly, S_ℓ is a finite set. We consider the following algorithm.

```
$\mathbb{A}_\ell$    // A Turing machine $\mathbb{M}$, $x \in \Sigma^*$, and $1^m$ with $m \in \mathbb{N}$

1. if $|\mathbb{M}| + |x| \leq \ell$
2.     then if there is a tuple $(\mathbb{M}, x, s) \in S_\ell$ and $s \leq 2^{\lfloor m/(|\mathbb{M}|+|x|) \rfloor}$
3.         then accept else reject
4.     else simulate $\mathbb{M}$ on $x$ at most $2^{\lfloor m/(|\mathbb{M}|+|x|) \rfloor}$ steps
5.     if the simulation accepts then accept
6.         else reject.
```

It is easy to see that the algorithm $\mathbb{A}_\ell$ decides Exp-Halt in time $O(2^{n/\ell})$.

Now we prove that Exp-Halt $\notin$ DTIME $\left(2^{o(n)}\right)$. Towards a contradiction, assume that $\mathbb{A}$ is an algorithm deciding Exp-Halt in time $2^{n/\iota(n)}$ for a nondecreasing and unbounded function $\iota : \mathbb{N} \to \mathbb{N}$. Let $\mathbb{M}_0$ be the following Turing machine:

```
$\mathbb{M}_0$    // $x \in \{0,1\}^*$

1. if $x$ is not the encoding of a Turing machine then reject
2. determine the Turing machine $\mathbb{M}$ with $x = \mathrm{enc}(\mathbb{M})$
3. $m \leftarrow \max\{12 \cdot |x|,$ number of steps performed by $\mathbb{M}_0$ so far$\}$
4. $s \leftarrow 2^{\lfloor m/(4 \cdot |x|) \rfloor}$
5. simulate at most $s$ steps of $\mathbb{A}$ on $(\mathbb{M}, x, 1^m)$
6. if the simulation does not halt, then $m \leftarrow m + 1$ and goto 4
7. if $\mathbb{A}$ accepts $(\mathbb{M}, x, 1^m)$ in at most $s$ steps then reject else accept.
```

We finish the proof by a diagonal argument: We set $x_0 := \mathrm{enc}(\mathbb{M}_0)$ and start $\mathbb{M}_0$ with input x_0. As ι is nondecreasing and unbounded, for sufficiently large $m \in \mathbb{N}$ we have

$$2^{(|\mathbb{M}_0|+|x_0|+m)/\iota(|\mathbb{M}_0|+|x_0|+m)} \leq 2^{\lfloor m/(4 \cdot |x_0|) \rfloor}.$$

As the left hand side of this inequality is an upper bound for the running time of the algorithm $\mathbb{A}$ on input $(\mathbb{M}_0, x_0, 1^{m_0})$, we see that eventually the Turing machine $\mathbb{M}_0$ with input x_0 reaches an m, we call it m_0, such that the simulation in Line 5 halts, more precisely,

$$\mathbb{A} \text{ halts on } (\mathbb{M}_0, x_0, 1^{m_0}) \text{ in at most } 2^{\lfloor m_0/(4 \cdot |x_0|) \rfloor} \text{ steps.} \tag{11}$$

At that point the number of steps (of the run of $\mathbb{M}_0$ on input x_0) is bounded by

$$\sum_{i \in [m_0]} 2 \cdot 2^{\lfloor i/(4 \cdot |x_0|) \rfloor} < 2^{\lfloor m_0/(4 \cdot |x_0|) \rfloor + 2}.$$

Hence, as $|\mathbb{M}_0| = |x_0|$ and $\lfloor m_0/(4 \cdot |x_0|) \rfloor \geq 3$ by Line 3, we get

$$\mathbb{M}_0 \text{ on } x_0 \text{ halts in} \leq 2 + 2^{\lfloor m_0/(4 \cdot |x_0|) \rfloor + 2} \leq 2^{\lfloor m_0/(|\mathbb{M}_0|+|x_0|) \rfloor} \text{ steps,} \tag{12}$$

Putting all together we get the desired contradiction:

$$\begin{aligned}
&\mathbb{M}_0 \text{ accepts } x_0 \\
\Longleftrightarrow\ &\mathbb{M}_0 \text{ accepts } x_0 \text{ in} \leq 2^{\lfloor m_0/(|\mathbb{M}_0|+|x_0|)\rfloor} \text{ steps} && \text{(by (12))} \\
\Longleftrightarrow\ &\mathbb{A} \text{ accepts } (\mathbb{M}_0, x_0, 1^{m_0}) && \text{(as } \mathbb{A} \text{ decides E}\textsc{xp}\text{-H}\textsc{alt}) \\
\Longleftrightarrow\ &\mathbb{A} \text{ accepts } (\mathbb{M}_0, x_0, 1^{m_0}) \text{ in at most } 2^{\lfloor m_0/(4\cdot|x_0|)\rfloor} \text{ steps} && \text{(by (11))} \\
\Longleftrightarrow\ &\mathbb{M}_0 \text{ rejects } x_0 && \text{(by Line 7 in the definition of } \mathbb{M}_0). \quad \square
\end{aligned}$$