# $W$-Hardness under Linear FPT-Reductions: Structural Properties and Further Applications*

Jianer Chen[1] Xiuzhen Huang[2] Iyad A. Kanj[3] Ge Xia[4]

[1] Dept. of Computer Science, Texas A&M University, College Station, TX 77843; College of Information Science and Engineering, Central South University, Changsha 410083, P.R.China. Email: `chen@cs.tamu.edu`

[2] Computer Science Department, Arkansas State University, State University, Arkansas 72467, USA. Email: `xzhuang@csm.astate.edu`

[3] School of Computer Science, Telecommunications and Information Systems, DePaul University, 243 S. Wabash Avenue, Chicago, IL 60604-2301, USA. Email: `ikanj@cs.depaul.edu`

[4] Department of Computer Science, Lafayette College, Easton, PA 18042, USA. Email: `gexia@cs.lafayette.edu`

**Abstract.** The notion of linear fpt-reductions has been recently used to derive strong computational lower bounds for well-known NP-hard problems. In this paper, we formally investigate the notions of $W[t]$-hardness and $W[t]$-completeness under the linear fpt-reduction, and study structural properties of the corresponding complexity classes. Additional complexity lower bounds on important computational problems are also established.

## 1 Introduction

A *parameterized problem* $Q$ is a decision problem consisting of instances of the form $(x, k)$, where the integer $k \geq 0$ is called the *parameter*. The parameterized problem $Q$ is *fixed-parameter tractable* [8] if it can be solved in time $f(k)|x|^{O(1)}$, where $f$ is a recursive function[5]. Certain NP-hard parameterized problems, such as VERTEX COVER, are fixed-parameter tractable, and hence can be solved practically for small parameter values [7]. On the other hand, the inherent computational difficulty of solving many other NP-hard parameterized problems with even small parameter values has suggested that certain parameterized problems be not fixed-parameter tractable, which has motivated the *theory of fixed-parameter intractability* [8]. The $W$-hierarchy $\bigcup_{t \geq 0} W[t]$ has been introduced to characterize the inherent level of intractability for parameterized problems. A large number of parameterized problems have been proved to be hard or complete for various levels in the $W$-hierarchy [8]. Examples of $W[1]$-hard problems

* This research is supported in part by US NSF under Grants CCR-0311590 and CCF-0430683, by China NNSF under Grants No.60373083 and No.60433020, and by DePaul University Competitive Research Grant

[5] In this paper, we always assume that complexity functions are "nice" with both domain and range being non-negative integers and the values of the functions and their inverses can be easily computed.

include many well-known NP-hard problems such as CLIQUE, DOMINATING SET, SET COVER, and WEIGHTED CNF SATISFIABILITY. The theory of parameterized intractability has found important applications in a variety of areas such as database systems and model checking [9, 10, 16].

The $W[1]$-hardness of a parameterized problem provides a strong evidence that the problem is not solvable in time $f(k)n^{O(1)}$ for any function $f$. However, $W[1]$-hardness dose not provide further information on how precisely the problem complexity depends on the parameter $k$. For example, the $W[1]$-hardness of the CLIQUE problem does not exclude the possibility of solving the problem in time $O(n^{\log\log k})$. Note that such an algorithm would be practically acceptable for moderate values of the parameter $k$, such as $k = 1000$.

Recent investigation has started along this line of research. In particular, the concept of *linear fpt-reduction* has been introduced to derive stronger computational lower bounds for well-known NP-hard parameterized problems [4, 5]. For example, based on the linear fpt-reduction, it has been shown that unless an unlikely collapse occurs in the parameterized complexity theory, any algorithm solving the CLIQUE problem takes time at least $n^{\Omega(k)}$. Note that this lower bound is asymptotically tight in the sense that the trivial algorithm that enumerates all subsets of $k$ vertices in a given graph to test the existence of a clique of size $k$ runs in time $O(n^k)$.

Therefore, the linear fpt-reduction has provided a powerful method for deriving strong computational lower bounds. In this paper, we formally investigate the concepts of $W[t]$-hardness and $W[t]$-completeness under the linear fpt-reduction, and systematically study the structural properties of the corresponding complexity classes. These complexity classes are defined based on generic complete problems under the linear fpt-reduction, instead of on computational models as for most traditional complexity classes. Therefore, it is natural to ask whether the familiar structural properties for traditional complexity hierarchies still hold true for the new parameterized complexity hierarchy. Moreover, the study of the structural properties of the new complexity classes has a direct impact on the applications of the theory to derive strong computational lower bounds.

We then illustrate the power of our techniques by deriving complexity lower bounds for further computational problems. We note that many of the fpt-reductions proposed in the literature are actually linear fpt-reductions or can be easily modified to become linear fpt-reductions. This enables us to quickly expand the list of computational problems with strong complexity lower bounds. We also study the parameterized complexity of the problems in the classes LOGNP and LOGSNP introduced by Papadimitriou and Yannakakis [15]. These problems are solvable in time $n^{O(\log n)}$ and therefore look "easier" than NP-hard problems in general. In particular, we study the problems TOURNAMENT DOMINATING SET, RICH HYPERGRAPH COVER, and V-C DIMENSION in these classes, and prove that, these problems are $W[1]$-hard under the linear fpt-reduction. In consequence, unless an unlikely collapse occurs in the parameterized complexity theory, these problems cannot be solved in time $f(k)n^{o(k)}$, neither can the optimization versions of these problems have polynomial time approximation schemes of running time $f(1/\epsilon)n^{o(1/\epsilon)}$, for any function $f$. These results either improve or complement previous research on the problems, and advance our understanding on the complexity of the problems.

We briefly review the related terminologies. Denote by $FPT$ the class of all fixed-parameter tractable problems. A circuit $C$ is a *$\Pi_t$-circuit* if its output gate is an AND gate and it has depth $t$. The *weight* of an assignment $\tau$ to a circuit is the number of variables assigned value 1 by $\tau$. The parameterized problem WEIGHTED SATISFIABILITY on $\Pi_t$-circuits, abbreviated WCS[$t$], is to determine for a given $\Pi_t$-circuit $C$ and an integer $k$, whether $C$ has a satisfying assignment of weight $k$. The WEIGHTED MONOTONE SATISFIABILITY (resp. WEIGHTED ANTIMONOTONE SATISFIABILITY) problem on $\Pi_t$-circuits, abbreviated WCS$^+[t]$ (resp. WCS$^-[t]$) is defined similarly as WCS[$t$] except that the circuit $C$ is required to be monotone (resp. antimonotone). To simplify our statements, we will denote by WCS$^*[t]$ the problem WCS$^+[t]$ if $t$ is even and the problem WCS$^-[t]$ if $t$ is odd. Finally, the WEIGHTED ANTIMONOTONE CNF 2SAT problem, abbreviated WCNF 2SAT$^-$, consists of the pairs of the form $(F, k)$, where $k$ is an integer $k$ and $F$ is a CNF formula in which all literals are negative and each clause contains at most 2 literals, such that $F$ has a satisfying assignment of weight $k$.

Due to the space limit, proofs for Theorems 1, 2, 3, and 5 have been omitted. We refer interested readers to the full version of the paper [6].

## 2 $W_l$-hardness and $W_l$-completeness

Each instance $(C, k)$ of the WCS[$t$] problem can be regarded as a search problem, in which we need to select $k$ elements from a search space consisting of a set of $n$ input variables, and assign them value 1 so that the circuit $C$ is satisfied. Many well-known computational problems, such as WEIGHTED CNF SAT, SET COVER, and HITTING SET, have similar formulations. The interested reader is referred to [5] for detailed discussion on this issue.

We will concentrate on parameterized problems that seek a subset in a search space satisfying certain properties. Thus, each instance of our parameterized problem is associated with a triple $(k, n, m)$, where $k$ is the parameter, $n$ is the size of the search space, and $m$ is the instance size [6].

**Definition 1.** A parameterized problem $Q$ is *linear fpt-reducible*, shortly *fpt$_l$-reducible*, to a parameterized problem $Q'$ if there exist a function $f$ and an algorithm $A$ of running time $f(k)m^{O(1)}$ that, on each $(k, n, m)$-instance $x$ of $Q$, produces a $(k', n', m')$-instance $x'$ of $Q'$, where $k' = O(k)$, $n' = n^{O(1)}$, $m' = m^{O(1)}$, and $x$ is a yes-instance of $Q$ if and only if $x'$ is a yes-instance of $Q'$.

It is easy to verify that the fpt$_l$-reducibility is transitive. Similar to the $W$-hierarchy defined in terms of the standard fpt-reducibility [8], we introduce a $W$-hierarchy based on the fpt$_l$-reducibility.

[6] For most problems in our consideration, the search space can be easily identified. For problems in which the search space is not easily identified, we simply let $n = m$.

**Definition 2.** A parameterized problem $Q$ is *$W[1]$-hard under the fpt$_l$-reduction*, shortly *$W_l[1]$-hard*, if the WCNF 2SAT$^-$ problem is fpt$_l$-reducible to $Q$. The problem $Q$ is *$W[t]$-hard under fpt$_l$-reduction*, shortly *$W_l[t]$-hard*, for $t \geq 2$ if the WCS$^*[t]$ problem is fpt$_l$-reducible to $Q$. For all $t \geq 1$, a parameterized problem $Q$ is *$W_l[t]$-complete* if $Q$ is in $W[t]$ and is $W_l[t]$-hard.

The $W_l[t]$-hardness has been used to derive strong complexity lower bounds. For $W_l[t]$-hard problems where $t \geq 2$, we have the following result.

**Proposition 1.** (Theorem 5.1, [5]) *For any integer $t \geq 2$, unless $W[t-1] = FPT$, no $W_l[t]$-hard problem can be solved in time $f(k)n^{o(k)}m^{O(1)}$ for any recursive function $f$.*

Computational lower bounds for $W_l[1]$-hard problems have been closely related to the *exponential time hypothesis* (ETH), which was first articulated in [12]. This hypothesis conjectures that the problem 3-SATISFIABILITY cannot be solved in time $2^{o(n)}$. To support the hypothesis, Impagliazzo and Paturi [12] have shown that if ETH fails then many well-known NP-hard problems, including all SNP problems formulated in [14], are solvable in subexponential time. Note that many of the SNP problems have been the targets for exact algorithms for decades but no subexponential time algorithms for them have been developed.

It is known [8] that ETH implies $W[1] \neq FPT$.

**Proposition 2.** (Theorem 5.2, [5]) *Unless ETH fails, no $W_l[1]$-hard problem is solvable in time $f(k)m^{o(k)}$ for any recursive function $f$.*

The main result in this section is that for any $t \geq 1$, $W_l[t+1]$-hardness implies $W_l[t]$-hardness. There are a number of reasons why this result is not trivial and should be examined carefully:

- In most hierarchies in complexity theory, the hardness for an upper level implies trivially the hardness for a lower level. For example, a $\Sigma^p_{t+1}$-hard problem in the polynomial time hierarchy is automatically $\Sigma^p_t$-hard by the definitions [13]. Therefore, it will be interesting to check whether such a common property is also shared by the $W_l$-hierarchy.
- Such a result does not trivially follow from the definitions. The $W_l[t]$-hardness is defined differently according to the parity of the integer $t$: for an even integer $t$, $W_l[t]$-hardness is defined based on the satisfiability problem WCS$^+[t]$ on monotone circuits, while for an odd integer $t$, $W_l[t]$-hardness is defined based on the satisfiability problem WCS$^-[t]$ on antimonotone circuits. In particular, the fpt-reduction from the problem WCS$^*[t-1]$ to the problem WCS$^*[t]$ proposed in the literature [8] is *not* a linear fpt-reduction (see Chapter 12 in [8] for details).
- Note that the lower bound in Proposition 2 is actually stronger than that in Proposition 1 since the search space size $n$ is in general not larger than the instance size $m$. That is, $W_l[1]$-hardness in fact implies a stronger lower bound (although also under a stronger working hypothesis) than that implied by $W_l[t]$-hardness for $t > 1$. Therefore, proving that $W_l[t]$-hardness implies

$W_l[t-1]$-hardness will immediately provide a stronger computational lower bound for $W_l[t]$-hard problems when $t > 1$.

**Theorem 1.** *For any $t \geq 2$, $W_l[t+1]$-hardness implies $W_l[t]$-hardness.*

**Theorem 2.** *$W_l[2]$-hardness implies $W_l[1]$-hardness.*

## 3 New lower bounds

Propositions 1 and 2 offer powerful techniques for deriving strong complexity lower bounds for well-known NP-hard problems. In particular, it has been shown [4, 5] that the following parameterized problems are $W_l[2]$-hard: WEIGHTED CNF SATISFIABILITY, SET COVER, HITTING SET, and DOMINATING SET, and that the following parameterized problems are $W_l[1]$-hard: WEIGHTED CNF $q$-SAT for any integer $q \geq 2$, CLIQUE, and INDEPENDENT SET. According to Proposition 2, none of these problems can be solved in time $f(k)m^{o(k)}$ for any recursive function $f$ unless ETH fails.

In this section we expand the list of $W_l[1]$-hard problems by developing linear fpt-reductions from the known $W_l[1]$-hard problems. In fact, many existing fpt-reductions proposed in the literature are linear fpt-reductions. Therefore, these fpt-reductions can be directly used or modified for our purpose. Using this approach, we can quickly get a much longer list of $W_l[1]$-hard problems and claim strong complexity lower bounds for these problems. The reader is referred to [8] for precise definitions for these problems.

**Theorem 3.** (1) *The following parameterized problems are $W_l[2]$-hard:* RED-BLUE DOMINATING SET*,* DOMINATING CLIQUE*,* PRECEDENCE CONSTRAINED PROCESSOR SCHEDULING*,* FEATURE SET*, and* WEIGHTED BINARY INTEGER PROGRAMMING*; and* (2) *The problem* SET PACKING *is $W_l[1]$-hard.*

*In particular, none of these problems can be solved in time $f(k)m^{o(k)}$ for any recursive function $f$ unless ETH fails.*

Again, Theorem 3 gives asymptotically tight complexity lower bounds in a very strong sense for these well-known NP-hard problems. For example, even though the DOMINATING CLIQUE problem can be trivially solved by exhaustive enumeration in time $O(n^k m)$ of all subsets of $k$ vertices, where $n$ is the number of vertices and $m$ is the instance size of the graph, solving the problem in time $f(k)m^{o(k)}$ is very unlikely for any recursive function $f$.

We further apply our technique to study two important problems in computational biology.

> LONGEST COMMON SUBSEQUENCE: given a set $S = \{s_1, s_2, ..., s_k\}$ of $k$ strings over a finite alphabet $\Sigma$, and an integer $\lambda > 0$, is there a string $s \in \Sigma^*$ of length $\lambda$, which is a subsequence of all of the $k$ strings in $S$? Here the parameter is $k$.

> SHORTEST COMMON SUPERSEQUENCE: given a set $S = \{s_1, s_2, ..., s_k\}$ of $k$ strings over a finite alphabet $\Sigma$, and an integer $\lambda > 0$, is there a string

$s \in \Sigma^*$ of length $\lambda$, which is a supersequence of all of the $k$ strings in $S$? Here the parameter is $k$.

**Theorem 4.** *The problems* LONGEST COMMON SUBSEQUENCE *and* SHORTEST COMMON SUPERSEQUENCE *are* $W_l[1]$*-hard. In consequence, they cannot be solved in time* $f(k)m^{o(k)}$ *for any function* $f$*, unless ETH fails.*

*Proof.* Pietrzak [17] has recently proved the $W[1]$-hardness for the problems LONGEST COMMON SUBSEQUENCE and SHORTEST COMMON SUPERSEQUENCE by fpt-reductions from CLIQUE. For LONGEST COMMON SUBSEQUENCE, Pietrzak developed a polynomial time algorithm $A_1$ that, on an instance $(G, k)$ of CLIQUE, produces an instance $(S_1, \lambda_1, k_1)$ for LONGEST COMMON SUBSEQUENCE, where $k_1 = k + 1$ and $|S_1| = O(k^8 n^7) = O(n^{15})$, such that $(G, k)$ is a yes-instance of CLIQUE if and only if $(S_1, \lambda_1, k_1)$ is a yes-instance of LONGEST COMMON SUBSEQUENCE. This fpt-reduction is obviously a linear fpt-reduction. In consequence, the LONGEST COMMON SUBSEQUENCE problem is $W_l[1]$-hard. Note that here we have simply let the search space size in an instance of LONGEST COMMON SUBSEQUENCE to be equal to the instance size.

The other polynomial time algorithm developed by Pietrzak transforms an instance $(G, k)$ of CLIQUE to an instance $(S_2, \lambda_2, k_2)$ for SHORTEST COMMON SUPERSEQUENCE, where $k_2 = k + 1$ and $|S_2| = O(k^8 n^7) = O(n^{15})$, which gives a linear fpt-reduction from CLIQUE to SHORTEST COMMON SUPERSEQUENCE. In consequence, SHORTEST COMMON SUPERSEQUENCE is $W_l[1]$-hard. □

To see the significance of Theorem 4, we quote a sentence from [17]:

> *Unless an unlikely collapse in the parameterized hierarchy occurs, this*[7] *rules out the existence of exact algorithms with running time* $f(k)n^{O(1)}$ *(i.e., exponential only in k) for those problems. This does not mean that there are no algorithms with much better asymptotic time-complexity than the known* $O(n^k)$ *algorithms based on dynamic programming, e.g., algorithms with running time* $n^{\sqrt{k}}$ *are not deemed impossible by our results.*

Therefore, Theorem 4 has strengthened the results in [17] significantly and advanced our understanding on the complexity of the problems: it is actually unlikely that the problems can be solved in time $n^{\gamma(k)}$ for *any* sublinear function $\gamma(k)$, and the known dynamic programming algorithms of running time $O(n^k)$ for the problems are actually asymptotically optimal.

## 4 On the complexity of LOGNP and LOGSNP problems

To further illustrate the power of our methods, we consider another group of computational problems introduced by Papadimitriou and Yannakakis [15].

[7] This refers to the results proved in [17] that the problems LONGEST COMMON SUBSEQUENCE and SHORTEST COMMON SUPERSEQUENCE are $W[1]$-hard.

A directed graph $G$ is a *tournament* if between each pair of vertices in $G$, there is exactly one directed edge. A hypergraph $H$ is a *rich hypergraph* if every edge in $H$ is incident on at least half of the vertices in $H$. In their study for the complexity classes LOGNP and LOGSNP, Papadimitriou and Yannakakis [15] have in particular considered the following problems:

> RICH HYPERGRAPH COVER: given a rich hypergraph $H = (V, E)$ and a parameter $k$, is there a subset $C$ of $k$ vertices in $H$ such that every edge in $H$ is incident on at least one vertex in $C$?
>
> TOURNAMENT DOMINATING SET: given a tournament graph $T$, and a parameter $k$, is there a subset $D$ of $k$ vertices in $T$ such that for each vertex $v$ not in $D$, there is at least one vertex $w$ in $D$ and $[w, v]$ is a directed edge in $T$?
>
> V-C DIMENSION: given a family $\mathcal{F}$ of subsets of a universe $U$, and a parameter $k$, is there a subset $S$ of $U$ such that $|S| = k$ and for each subset $T$ of $S$, there is a set $C_T \in \mathcal{F}$ satisfying $S \cap C_T = T$?

It can be shown [15] that if the parameter value $k$ is larger than $\log m$, where $m$ is the instance size, then the answer to RICH HYPERGRAPH COVER and TOURNAMENT DOMINATING SET is always positive while the answer to V-C DIMENSION is always negative. Therefore, the problem instances of these problems become non-trivial only when $k \leq \log m$. In consequence, all these problems can be solved in time $O(m^{\log m})$. Hence, these problems are unlikely to be NP-hard. On the other hand, it is unknown whether any of these problems is solvable in polynomial time.

**Theorem 5.** *The problems* RICH HYPERGRAPH COVER *and* TOURNAMENT DOMINATING SET *are $W_l[2]$-hard, and the problem* V-C DIMENSION *is $W_l[1]$-hard. In consequence, they cannot be solved in time $f(k)m^{o(k)}$ for any function $f$, unless ETH fails.*

The approximability of the problems in Theorem 5 has drawn research interests recently [1, 2]. Recall that an NP optimization problem $Q$ has a *polynomial time approximation scheme* if there is an approximation algorithm $A_Q$ that takes a pair $(x, \epsilon)$ as input, where $x$ is an instance of $Q$ and $\epsilon > 0$ is a real number, and returns a solution $y$ for $x$ such that the approximation ratio of the solution $y$ is bounded by $1 + \epsilon$, and for a fixed $\epsilon > 0$, the running time of the algorithm $A_Q$ is bounded by a polynomial of $|x|$. The algorithm $A_Q$ is a *fully polynomial time approximation scheme* for $Q$ if the running time of $A_Q$ is bounded by a polynomial of $1/\epsilon$ and $|x|$, and is an *efficient polynomial time approximation scheme* for $Q$ [3] if the running time of $Q$ is bounded by $f(1/\epsilon)|x|^{O(1)}$ for a function $f$.

An NP optimization problem $Q$ can be systematically parameterized [5] into a parameterized problem *Para*($Q$), whose instances take the form $(x, k)$ asking whether the optimal value for $x$ is not larger than $k$ (resp. not smaller than $k$) in case $Q$ is a minimization (resp. maximization) problem.

**Theorem 6.** (Theorem 6.1, [5]) *If the parameterized version Para($Q$) of an NP optimization problem $Q$ is $W_l[1]$-hard, then $Q$ has no polynomial time approximation scheme of running time $f(1/\epsilon)m^{o(1/\epsilon)}$ for any recursive function $f$, unless ETH fails.*

Consider the following optimization problems.

> RICH HYPERGRAPH COVER-OPT: given a rich hypergraph $H = (V, E)$, find a minimum set $C$ of vertices such that each edge in $H$ is incident on at least one vertex in $C$.
>
> TOURNAMENT DOMINATING SET-OPT: given a tournament graph $T$, find a minimum set $D$ of vertices such that for each vertex $v$ not in $D$, there is at least one vertex $w \in D$ and $[w, v]$ is a directed edge in $T$.
>
> V-C DIMENSION-OPT: given a family $\mathcal{F}$ of subsets of a universe $U$, find a maximum subset $S$ of $U$ such that for each subset $T$ of $S$, there is a set $C_T \in \mathcal{F}$ satisfying $S \cap C_T = T$.

**Theorem 7.** *Unless ETH fails, none of the problems* RICH HYPERGRAPH COVER-OPT*,* TOURNAMENT DOMINATING SET-OPT*, and* V-C DIMENSION-OPT *has polynomial time approximation schemes of running time $f(1/\epsilon)m^{o(1/\epsilon)}$ for any recursive function $f$.*

*Proof.* The parameterized versions of these problems are just the corresponding parameterized problems in Theorem 5. The theorem follows immediately from Theorem 5 and Theorem 6. □

Theorem 7 improves or complements a number of previous results. Papadimitriou and Yannakakis [15] introduced the classes LOGNP and LOGSNP, and proved that RICH HYPERGRAPH COVER and TOURNAMENT DOMINATING SET are complete under the polynomial time reduction for the class LOGSNP, and that V-C DIMENSION is complete under the polynomial time reduction for the class LOGNP. These results hint that it is unlikely that these problems can be solved in polynomial time. Theorem 7 shows that these problems are not only difficult for being solved precisely in polynomial time, but also difficult for being solved approximately in polynomial time. Cai and Chen [1] showed that the parameterized version of every NP optimization problem with fully polynomial time approximation schemes is fixed-parameter tractable, and Cesati and Trevisan [3] extended this result and proved that the parameterized version of every NP optimization problem with efficient polynomial time approximation schemes is fixed-parameter tractable. As a consequence, these results plus the $W[1]$-hardness of the problems RICH HYPERGRAPH COVER, TOURNAMENT DOMINATING SET, and V-C DIMENSION imply that these problems have no fully or efficient polynomial time approximation schemes. Theorem 7 further strengthens these results by showing the impossibility for these problems to have polynomial time approximation schemes of running time $f(1/\epsilon)m^{o(1/\epsilon)}$ for any recursive function $f$. Cai, Juedes, and Kanj [2] studied the approximability of these problems and proved that TOURNAMENT DOMINATING SET and RICH HYPERGRAPH COVER cannot be approximated to a ratio $c > 1$ unless DOMINATING SET can be approximated to a ratio $2c$ in time $O(2^{n^\delta})$ for some $\delta < 1$. Theorem 7 complements this result by showing the inpracticability of approximation algorithms with small approximation ratio for TOURNAMENT DOMINATING SET and RICH HYPERGRAPH COVER (under a different working hypothesis). Moreover, it was posed as an open problem in [2] to study the inapproximability of V-C DIMENSION, and Theorem 7 provides an answer to this question.

We point out that the results in Theorems 5 and 7 can be extended to many other problems, such as the problems LOG CLIQUE, LOG DOMINATING SET, LOG HYPERGRAPH COVER, LOG ADJUSTMENT, LOG CHORDLESS PATH studied in [15]. For detailed discussions, interested readers are referred to [11].

## 5 A remark on the *W*-hierarchy

In this section, we provide an interesting observation on the original $W$-hierarchy. Most complexity hierarchies have the “hierarchical collapsing property” so that the collapsing of a lower level in the hierarchy implies the collapsing of all higher levels. For example, if for any integer $t > 0$, the $t$-th level of the polynomial time hierarchy collapses to the $(t-1)$-st level, $\Sigma_t^p = \Sigma_{t-1}^p$, then the entire polynomial time hierarchy collapses to the $(t-1)$-st level: $\Sigma_{t+h}^p = \Sigma_{t-1}^p$ for all $h \geq 0$ [13]. Most important complexity hierarchies, such as the NC hierarchy, the AC hierarchy, and the Boolean hierarchy, share a similar collapsing result [13].

It has been a well-known open problem in parameterized complexity theory whether the $W$-hierarchy satisfies a similar collapsing result. In particular, we are interested in knowing whether the following result holds true for the $W$-hierarchy:

**Collapsing.** If $W[t] = FPT$ for an integer $t \geq 1$, then $W[h] = FPT$ for all integer $h \geq t$.

One would expect naturally that the collapsing results such as **Collapsing** hold true. In the following, we discuss the consequence of **Collapsing**.

**Theorem 8.** *If* **Collapsing** *holds true, then the problem* WCS$^*[t]$ *either cannot be solved in time $f_1(k)n^{o(k)}m^{O(1)}$ for any function $f_1$, or can be solved in time $f_2(k)m^{O(1)}$ for a fixed function $f_2$.*

*Proof.* Suppose that the problem WCS$^*[t]$ can be solved in time $f_1(k)n^{o(k)}m^{O(1)}$ for a function $f_1$. By Proposition 1, this implies that $W[t-1] = FPT$. By **collapsing**, this would imply $W[t] = FPT$. Since WCS$^*[t]$ is in $W[t]$, we derive that WCS$^*[t]$ can be solved in time $f_2(k)m^{O(1)}$ for a function $f_2$. □

Obviously, the problem WCS$^*[t]$ in Theorem 8 can be replaced by any $W_l[t]$-complete problem.

Note that if **Collapsing** can be proved, then the conclusion in Theorem 8 holds true *unconditionally*, not depending on any complexity assumptions such as P $\neq$ NP or $W[1] \neq FPT$. This would exclude the possibility that, for example, the complexity of the CLIQUE problem is in the order of $\Theta(n^{\sqrt{k}})$.

## References


1. L. Cai and J. Chen, On fixed parameter tractability and approximability of NP optimization problems, *Journal of Computer and System Sciences 54*, pp. 465-474, (1997).
2. L. Cai, D. Juedes, and I. A. Kanj, Inapproximability of non NP-hard optimization problems, *Theoretical Computer Science 289*, pp. 553-571, (2002).
3. M. Cesati and L. Trevisan, On the efficiency of polynomial time approximation schemes, *Information Processing Letters 64*, pp. 165-171, (1997).
4. J. Chen, B. Chor, M. Fellows, X. Huang, D. Juedes, I. Kanj, and G. Xia, Tight lower bounds for certain parameterized NP-hard problems, *Proc. 19th Annual IEEE Conference on Computational Complexity* (CCC 2004), pp. 150-160, (2004). Journal version is to appear in *Information and Computation.*
5. J. Chen, X. Huang, I. Kanj, and G. Xia, Linear FPT reductions and computational lower bounds, *Proc. 36th Annual ACM Symposium on Theory of Computing* (STOC 2004), pp. 212-221, (2004).
6. J. Chen, X. Huang, I. Kanj, and G. Xia, $W$-hardness under linear FPT-reductions: structural properties and further applications, *Tech. Report*, Dept. Computer Science, Texas A&M University, (2005).
7. J. Chen, I. A. Kanj, and W. Jia, Vertex cover: further observations and further improvements, *Journal of Algorithms 41*, pp. 280-301, (2001).
8. R.G. Downey and M.R. Fellows, *Parameterized Complexity*, Springer-Verlag, 1999.
9. J. Flum and M. Grohe, Model-checking problems as a basis for parameterized intractability, *Proc. 19th IEEE Symposium on Logic in Computer Science*, (LICS'04), pp. 388-397, (2004).
10. M. Grohe, The parameterized complexity of database queries, *Proc. 20th ACM Symposium on Principles of Database Systems*, (PODS'01), pp. 82-92, (2001).
11. X. Huang, *Parameterized Complexity and Polynomial-time Approximation Schemes*, Ph.D. Dissertation, Department of Computer Science, Texas A&M University, December, 2004.
12. R. Impagliazzo and R. Paturi, Which problems have strongly exponential complexity? *Journal of Computer and System Sciences 63*, pp. 512-530, (2001).
13. C.H. Papadimitriou, *Computational Complexity*, Addison-Wesley Pub., Reading, Mass., 1995.
14. C. H. Papadimitriou and M. Yannakakis, Optimization, approximation, and complexity classes, *Journal of Computer and System Sciences 43*, pp. 425-440, (1991).
15. C.H. Papadimitriou and M. Yannakakis, On limited nondeterminism and the complexity of VC dimension, *Journal of Computer and System Sciences 53*, pp. 161-170, (1996).
16. C.H. Papadimitriou and M. Yannakakis, On the complexity of database queries, *Journal of Computer and System Sciences 58*, pp. 407-427, (1999).
17. K. Pietrzak, On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems, *Journal of Computer and System Sciences 67*, pp. 757-771, (2003).