Information Processing Letters 132 (2018) 1–5

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

Why is it hard to beat $O(n^2)$ for Longest Common Weakly Increasing Subsequence?

Adam Polak [1]

Department of Theoretical Computer Science, Faculty of Mathematics and Computer Science, Jagiellonian University, Łojasiewicza 6, 30-348 Kraków, Poland

ARTICLE INFO

Article history:
Received 24 May 2017
Received in revised form 21 November 2017
Accepted 21 November 2017
Available online 2 December 2017
Communicated by Marcin Pilipczuk

Keywords:
Computational complexity
Longest common weakly increasing subsequence
Lower bound
SETH

ABSTRACT

The Longest Common Weakly Increasing Subsequence problem (LCWIS) is a variant of the classic Longest Common Subsequence problem (LCS). Both problems can be solved with simple quadratic time algorithms. A recent line of research led to a number of matching conditional lower bounds for LCS and other related problems. However, the status of LCWIS remained open.

In this paper we show that LCWIS cannot be solved in $O(n^{2-\varepsilon})$ time unless the Strong Exponential Time Hypothesis (SETH) is false.

The ideas which we developed can also be used to obtain a lower bound based on a safer assumption of NC-SETH, i.e. a version of SETH which talks about NC circuits instead of less expressive CNF formulas.

1. Introduction

Despite attracting interest of many researches, both from theoretical computer science and computational biology communities, for many years the classic Longest Common Subsequence problem (LCS) has not seen any significant improvement over the simple $O(n^2)$ dynamic programming algorithm. The current fastest, $O(n^2/\log^2 n)$ algorithm by Masek and Paterson [1], dates back to 1980.

Difficulties in making progress on the LCS inspired studying numerous related problems, among them the Longest Common Increasing Subsequence problem (LCIS), for which Yang, Huang, and Chao [2] found a quadratic time dynamic programming algorithm. Their algorithm was later improved by Sakai [3] to work in linear space. Even though both these algorithms are devised to compute the Longest Common Increasing Subsequence, they can be easily modified to compute the Longest Common Weakly Increasing Subsequence (LCWIS). The latter problem, first introduced by Kutz et al. [4], can be solved in linear time in the special case of a 3-letter alphabet, as proposed by Duraj [5]. However, despite some attempts over the last decade, no subquadratic time algorithm has been found for the general case of LCWIS.

A recent line of research led to a number of conditional lower bounds for polynomial time solvable problems. In particular Abboud, Backurs, and Vassilevska Williams [6], and independently Bringmann and Künnemann [7] proved that LCS cannot be solved in $O(n^{2-\varepsilon})$ time unless the Strong Exponential Time Hypothesis (SETH) is false.

Hypothesis 1 *(Strong Exponential Time Hypothesis). There is no $\varepsilon > 0$ such that for all $k \geqslant 3$, k-SAT on N variables can be solved in $O(2^{(1-\varepsilon)N})$ time.*

Moreover, Bringmann and Künnemann [7] proposed a general framework for proving quadratic time hardness of sequence similarity measures. Within this framework, it is sufficient to show that a similarity measure *admits an align-*

E-mail address: polak@tcs.uj.edu.pl.

[1] This work was supported by the Polish Ministry of Science and Higher Education program *Diamentowy Grant* under grant no. DI2012 018942.

https://doi.org/10.1016/j.ipl.2017.11.007
0020-0190/

ment gadget to prove that this similarity measure cannot be computed in $O(n^{2-\varepsilon})$ time unless SETH is false. Besides LCS many other similarity measures, e.g. Edit Distance and Dynamic Time Warping, fall into this framework. However, it seems that neither LCIS nor LCWIS admits an alignment gadget.

In this paper we show that LCWIS cannot be solved in $O(n^{2-\varepsilon})$ time unless SETH is false.

Theorem 2. *If the Longest Common Weakly Increasing Subsequence problem for two sequences of length n can be solved in $O(n^{2-\varepsilon})$ time, then given a CNF formula on N variables and M clauses it is possible to compute the maximum number of satisfiable clauses (MAX-CNF-SAT) in $O(2^{(1-\varepsilon/2)N}\operatorname{poly}(M))$ time.*

Our reduction is modeled after previous hardness results based on SETH, in particular [6] and [8]. We go through the Most-Orthogonal Vectors problem, and construct *vector gadgets* such that two vector gadgets have large LCWIS if and only if the corresponding vectors have small inner product. The crucial ingredient, presented in Lemma 9, is a construction that lets us combine many vector gadgets into two sequences such that their LCWIS depends on the largest LCWIS among all pairs of vector gadgets.

Unfortunately, our uncomplicated techniques are not sufficient to prove similar lower bounds neither for LCIS nor for the generalization of LCWIS to more than two sequences. Recently, a more involved construction has been proposed to establish tight lower bounds for both these problems [9].

Unlike $\mathsf{P} \neq \mathsf{NP}$ and several other common assumptions for conditional lower bounds in computational complexity, SETH is considered by many not a very safe working hypothesis. Recently, Abboud et al. [10] came up with a weaker assumption, which still allows to prove many previous SETH-based lower bounds. More specifically, they propose a reduction from satisfiability of *Branching Programs* [11] (BP-SAT) to LCS (and, in general, any other similarity measure which admits an alignment gadget). Their reduction implies that the existence of a strongly subquadratic time algorithm for LCS would have much more remarkable consequences in computational complexity than just refuting SETH, e.g. an exponential improvement over brute-force algorithm for satisfiability of NC circuits. For an in-depth discussion of consequences of their reduction and motivations to study such reductions please refer to the original paper [10]. Their main development, which makes the reduction possible, is the construction of *reachability gadgets*, which encode computations of Branching Programs in the language of LCS and play a role analogous to vector gadgets in previous reductions from CNF-SAT via Orthogonal Vectors problem. It is easy to devise similar reachability gadgets for LCWIS, by adapting the original construction. Then, our Lemma 9 can be applied to these reachability gadgets to obtain a reduction from BP-SAT to LCWIS, giving even stronger evidence of the quadratic time hardness of LCWIS.

2. Preliminaries

Let us start with the formal definition of the LCWIS problem.

Definition 3 *(Longest Common Weakly Increasing Subsequence). Given two sequences A and B over an alphabet Σ with a linear order $\leqslant_\Sigma$, the* Longest Common Weakly Increasing Subsequence *problem asks to find a sequence C such that*

- *it is weakly increasing with respect to $\leqslant_\Sigma$,*
- *it is a subsequence of both A and B,*
- *and its length is maximum possible.*

We denote the length of C by $\mathrm{LCWIS}(A, B)$.

For example, $\mathrm{LCWIS}(\langle 1,2,5,2,5,3\rangle, \langle 2,4,5,2,3,4\rangle) = 3$, and the optimal subsequence is $\langle 2,2,3\rangle$.

To simplify further arguments we introduce, as an auxiliary problem, the weighted version of LCWIS.

Definition 4 *(Weighted Longest Common Weakly Increasing Subsequence). Given two sequences A and B over an alphabet Σ with a linear order $\leqslant_\Sigma$ and the* weight function $w : \Sigma \longrightarrow \mathbb{N}_+$, *the* Weighted Longest Common Weakly Increasing Subsequence *problem (WLCWIS) asks to find a sequence C such that*

- *it is weakly increasing with respect to $\leqslant_\Sigma$,*
- *it is a subsequence of both A and B,*
- *and its total weight, i.e. $\sum_{i=1}^{|C|} w(C_i)$, is maximum possible.*

We denote the total weight of C by $\mathrm{WLCWIS}(A, B)$.

Lemma 5. *Computing the* WLCWIS *of two sequences, each of total weight at most W, can be reduced to computing the* LCWIS *of two sequences, each of length at most W.*

Proof. For a sequence $X = \langle X_1, X_2, \ldots, X_{|X|}\rangle$ let $\widehat{X}$ denote a sequence obtained from X by replacing each symbol a by its $w(a)$ many copies, i.e.

$$\widehat{X} = X_1^{w(X_1)}\, X_2^{w(X_2)}\, \ldots\, X_{|X|}^{w(X_{|X|})}.$$

We will show that $\mathrm{WLCWIS}(A, B) = \mathrm{LCWIS}(\widehat{A}, \widehat{B})$. Every common weakly increasing subsequence C of A and B translates to a common weakly increasing subsequence $\widehat{C}$ of $\widehat{A}$ and $\widehat{B}$, and the length of $\widehat{C}$ equals the total weight of C, thus $\mathrm{WLCWIS}(A, B) \leqslant \mathrm{LCWIS}(\widehat{A}, \widehat{B})$.

It remains to prove $\mathrm{WLCWIS}(A, B) \geqslant \mathrm{LCWIS}(\widehat{A}, \widehat{B})$. Let $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_{|\Sigma|}\}$, $\sigma_i <_\Sigma \sigma_{i+1}$. Let us represent the longest common weakly increasing subsequence of $\widehat{A}$ and $\widehat{B}$ as $\sigma_1^{\alpha_1}\sigma_2^{\alpha_2}\ldots\sigma_{|\Sigma|}^{\alpha_{|\Sigma|}}$. Note that such a representation is possible because the subsequence is weakly increasing. Let

$$C := \sigma_1^{\lceil \alpha_1/w(\sigma_1)\rceil}\sigma_2^{\lceil \alpha_2/w(\sigma_2)\rceil}\ldots\sigma_{|\Sigma|}^{\lceil \alpha_{|\Sigma|}/w(\sigma_{|\Sigma|})\rceil}.$$

Note that the total weight of C with respect to w is at least $\mathrm{LCWIS}(\widehat{A}, \widehat{B})$. To finish the proof observe that C is a subsequence of both A and B. Indeed, C is a subsequence of A because, for each i, α_i occurrences of σ_i in $\widehat{A}$ must originate from at least $\lceil \alpha_i/w(\sigma_i)\rceil$ different occurrences of σ_i in