Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Efficient algorithms for shortest partial seeds in words

Tomasz Kociumaka [a], Solon P. Pissis [b], Jakub Radoszewski [a,b,*], Wojciech Rytter [a], Tomasz Waleń [a]

[a] *Institute of Informatics, University of Warsaw, Warsaw, Poland*
[b] *Department of Informatics, King's College London, London, UK*

ARTICLE INFO

*Article history:*
Received 15 June 2016
Received in revised form 15 November 2016
Accepted 28 November 2016
Available online xxxx

*Keywords:*
Algorithms on strings
Quasiperiodicity
Suffix tree
Cover
Seed

ABSTRACT

A factor $u$ of a word $w$ is a *cover* of $w$ if every position in $w$ lies within some occurrence of $u$ in $w$. A factor $u$ is a *seed* of $w$ if it is a cover of a superstring of $w$. Covers and seeds extend the classical notions of periodicity. We introduce a new notion of *$\alpha$-partial seed*, that is, a factor covering as a seed at least $\alpha$ positions in a given word. We use the Cover Suffix Tree, recently introduced in the context of *$\alpha$-partial covers* (Kociumaka et al., 2015, [13]); an $\mathcal{O}(n \log n)$-time algorithm constructing such a tree is known. However, it appears that partial seeds are more complicated than partial covers—our algorithms require algebraic manipulations of special functions related to edges of the modified Cover Suffix Tree and the border array. We present a procedure for computing shortest $\alpha$-partial seeds that works in $\mathcal{O}(n)$ time if the Cover Suffix Tree is already given.

This is a full version, which includes all the proofs, of a paper that appeared at CPM 2014 [1].

## 1. Introduction

Periodicity in words is a fundamental topic in combinatorics on words and string algorithms (see, e.g., [2]). The concept of quasiperiodicity generalizes the notion of periodicity [3], and it allows detecting repetitive structure of words which cannot be found using the classic characterizations of periods. Several types of quasiperiods have been introduced; each of them reveals slightly different kind of repetitive structures.

The best-known type of quasiperiodicity is the *cover* of a word. A factor $u$ of a word $w$ is said to be a cover of $w$ if every position in $w$ lies within some occurrence of $u$ in $w$; we also say that $w$ is covered by $u$. An extension of the notion of a cover is that of a *seed*. In this case, the positions considered to be covered by a factor $u$ also include those within overhanging occurrences of $u$. Equivalently, a factor $u$ is a seed of $w$ if and only if $w$ is a factor of a word $y$ covered by $u$.

Several efficient algorithms for computation of covers and seeds have been developed since the introduction of these notions in early 1990s. The first one, by Apostolico et al. [4], is a linear-time procedure finding the shortest cover of a word. Moore and Smyth [5,6] showed that all covers can be determined in the same time complexity. Linear-time algorithms providing yet more complete characterizations of covers by so-called cover arrays were given in [7,8]. Seeds were introduced by Iliopoulos, Moore, and Park [9], who presented an $\mathcal{O}(n \log n)$-time algorithm identifying those factors. This result was much later improved by Kociumaka et al. [10], who gave a complex linear-time algorithm.

[*] Corresponding author.
*E-mail addresses:* kociumaka@mimuw.edu.pl (T. Kociumaka), solon.pissis@kcl.ac.uk (S.P. Pissis), jrad@mimuw.edu.pl (J. Radoszewski), rytter@mimuw.edu.pl (W. Rytter), walen@mimuw.edu.pl (T. Waleń).

http://dx.doi.org/10.1016/j.tcs.2016.11.035
0304-3975/© 2016 Elsevier B.V. All rights reserved.

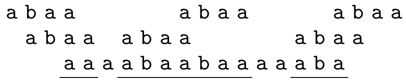

**Fig. 1.** The positions covered by abaa as a partial seed of $w$ are underlined. The word abaa is a 12-partial seed of $w$; it has four overhanging occurrences and two full occurrences. Note that a is the shortest 12-partial seed of $w$.

Even though quasiperiodicities give much more flexibility than the classic periodic structures, it remains unlikely that an arbitrary word has a cover or a seed other than the whole word itself. Due to this reason, relaxed variants of quasiperiodicity have been introduced. One of the ideas, resulting in the notions of *approximate covers* [11] and *approximate seeds* [12], is to require that each position lies within an approximate occurrence of the corresponding quasiperiod. Another approach, introduced recently in [13], yields the notion of *partial covers*—factors required to cover a certain number of positions of a word. Partial covers generalize the earlier notion of *enhanced covers* [14], which are additionally required to be borders (both prefixes and suffixes) of the word. In this paper, we extend the ideas behind partial covers and introduce the concept of a *partial seed*.

The *cover index* $\mathcal{C}(u, w)$ of a factor $u$ in a word $w$ has been defined in [13] as the number of positions in $w$ covered by (full) occurrences of $u$ in $w$. The word $u$ is called an *$\alpha$-partial cover* of $w$ if $\mathcal{C}(u, w) \geq \alpha$.

We call a non-empty prefix of $w$ that is also a proper suffix of $u$ a *left-overhanging* occurrence of $u$ in $w$. Symmetrically, a non-empty suffix of $w$ which is a proper prefix of $u$ is called a *right-overhanging* occurrence. The *seed index* of $u$ in $w$, denoted as $\mathcal{S}(u, w)$, is the number of positions in $w$ covered by full, left-overhanging, or right-overhanging occurrences of $u$ in $w$. We say that $u$ is an *$\alpha$-partial seed* of $w$ if $\mathcal{S}(u, w) \geq \alpha$. If the word $w$ is clear from the context, we use the simpler notation $\mathcal{C}(u)$ and $\mathcal{S}(u)$ instead of $\mathcal{C}(u, w)$ and $\mathcal{S}(u, w)$, respectively.

**Example 1.** For $w = \texttt{aaaabaabaaaaaba}$, see also Fig. 1, the seed indices of sample factors are as follows:

$$\mathcal{S}(\texttt{abaa}) = 12,\ \mathcal{S}(\texttt{aba}) = 10,\ \mathcal{S}(\texttt{ab}) = 7,\ \mathcal{S}(\texttt{a}) = 12.$$

We study the following two related problems:

PartialSeeds
**Input:** a word $w$ of length $n$ and a positive integer $\alpha \leq n$
**Output:** all shortest factors $u$ such that $\mathcal{S}(u, w) \geq \alpha$

LimitedLengthPartialSeeds
**Input:** a word $w$ of length $n$ and an interval $[\ell, r]$
**Output:** a factor $u$, $|u| \in [\ell, r]$, which maximizes $\mathcal{S}(u, w)$

In [13] a data structure called the *Cover Suffix Tree* and denoted by $CST(w)$ was introduced. For a word $w$ of length $n$ the size of $CST(w)$ is $\mathcal{O}(n)$ and the construction time is $\mathcal{O}(n \log n)$.

**Our results.** In this article, we extend the Cover Suffix Tree to support queries concerning partial seeds.

**Theorem 2.** *Given $CST(w)$, the* LimitedLengthPartialSeeds *problem can be solved in $\mathcal{O}(n)$ time.*

By applying binary search, Theorem 2 implies an $\mathcal{O}(n \log n)$-time solution to the PartialSeeds problem. However, the running time can be improved to $\mathcal{O}(n)$, provided that $CST(w)$ is given in advance.

**Theorem 3.** *Given $CST(w)$, the* PartialSeeds *problem can be solved in $\mathcal{O}(n)$ time.*

Our solution for the PartialSeeds problem can also recover *all* the shortest factors $u$ of $w$ that satisfy the condition $\mathcal{S}(u, w) \geq \alpha$.

**Structure of the paper.** In Section 2, we introduce basic notation related to words and suffix trees, and we recall the Cover Suffix Tree (*CST*). Next, in Section 3, we extend *CST* to obtain its counterpart suitable for computation of partial seeds, which we call the *Seed Suffix Tree* (*SST*). In Section 4, we introduce two abstract problems formulated in terms of simple functions which encapsulate the intrinsic difficulty of the PartialSeeds and LimitedLengthPartialSeeds problems. Solutions to these problems, presented in the following Section 5, essentially constitute the most involved part of our contribution. We summarize our results in the Conclusions section.

A preliminary version of this paper, with some proofs omitted due to space restrictions, appeared at CPM 2014 [1].

Please cite this article in press as: T. Kociumaka et al., Efficient algorithms for shortest partial seeds in words, Theoret. Comput. Sci. (2016), http://dx.doi.org/10.1016/j.tcs.2016.11.035