Parameterized Complexity of Superstring Problems

Ivan Bliznets[2], Fedor V. Fomin[1,2], Petr A. Golovach[1,2](✉), Nikolay Karpov[2], Alexander S. Kulikov[2], and Saket Saurabh[1,3]

[1] Department of Informatics, University of Bergen, Bergen, Norway
[2] Steklov Institute of Mathematics at St. Petersburg,
Russian Academy of Sciences, St. Petersburg, Russia
petr.golovach@ii.uib.no
[3] Institute of Mathematical Sciences, Chennai, India

Abstract. In the Shortest Superstring problem we are given a set of strings $S = \{s_1, \ldots, s_n\}$ and an integer ℓ and the question is to decide whether there is a superstring s of length at most ℓ containing all strings of S as substrings. We obtain several parameterized algorithms and complexity results for this problem.

In particular, we give an algorithm which in time $2^{O(k)}\,\mathrm{poly}(n)$ finds a superstring of length at most ℓ containing at least k strings of S. We complement this by the lower bound showing that such a parameterization does not admit a polynomial kernel up to some complexity assumption. We also obtain several results about "below guaranteed values" parameterization of the problem. We show that parameterization by compression admits a polynomial kernel while parameterization "below matching" is hard.

1 Introduction

We consider the Shortest Superstring problem defined as follows:

Shortest Superstring
Input: A set of n strings $S = \{s_1, \ldots, s_n\}$ over an alphabet Σ and a non-negative integer ℓ.
Question: Is there a string s of length at most ℓ containing all strings from S as substrings?

This is a well-known NP-complete problem [10] with a range of practical applications from DNA assembly [7] to data compression [9]. Due to this fact approximation algorithms for it are widely studied. The currently best known approximation guarantee $2\frac{11}{23}$ is due to Mucha [17]. At the same time the best known exact algorithms run in roughly 2^n steps and are known for more than

The research leading to these results has received funding from the Government of the Russian Federation (grant 14.Z50.31.0030) and Grant of the President of Russian Federation (MK-6550.2015.1).

F. Cicalese et al. (Eds.): CPM 2015, LNCS 9133, pp. 89–99, 2015.
DOI: 10.1007/978-3-319-19929-0_8

50 years already. More precisely, using known algorithms for the Traveling Salesman problem, Shortest Superstring can be solved either in time $O^*(2^n)$ and the same space by dynamic programming over subsets [3,13] or in time $O^*(2^n)$ and only polynomial space by inclusion-exclusion [14,16] (here, $O^*(\cdot)$ hides factors that are polynomial in the input length, i.e., $\sum_{i=1}^{n}|s_i|$). Such algorithms can only be used in practice to solve instances of very moderate size. Stronger upper bounds are known for a special case when input strings have bounded length [11,12]. There are heuristic methods for solving Traveling Salesman, and hence also Shortest Superstring, they are efficient in practice, however have no efficient provable guarantee on the running time (see, e.g., [1]).

In this paper, we study the Shortest Superstring problem from the parameterized complexity point of view. This field studies the complexity of computational problems with respect not only to input size, but also to some additional parameters and tries to identify parameters of input instances that make the problem tractable. Interestingly, prior to our work, except observations following from the known reductions to Traveling Salesman, not much about the parameterized complexity of Shortest Superstring was known. We refer to the survey of Bulteau et al. [4] for a nice overview of known results on parameterized algorithms and complexity of strings problems. Thus our work can be seen as the first non-trivial step towards the study of this interesting and important problem from the perspective of parameterized complexity.

Our Results. In this paper we study two types of parameterization for Shortest Superstring and present two kind of results. The first set of results concerns "natural" parameterization of the problem. We consider the following generalization of Shortest Superstring:

Partial Superstring
Input: A collection (multiset) of strings S over an alphabet Σ, and non-negative integers k, ℓ.
Question: Is there a string s of length at most ℓ such that s is a superstring of a collection of at least k strings $S' \subseteq S$?

If $k = |S|$, then this is Shortest Superstring. Notice that S can contain copies of the same string and a string of S can be a substring of another string of the collection. For Shortest Superstring, such cases could be easily avoided, but for Partial Superstring it is natural to assume that we have such possibilities.

Here we show that Partial Superstring is fixed parameter tractable (FPT) when parameterized by k or ℓ. We complement this result by showing that it is unlikely that the problem admits a polynomial kernel with respect to these parameters.

The second set of results concerns "below guaranteed value" parameterization. Note that an obvious (non-optimal) superstring of $S = \{s_1, \ldots, s_n\}$ is a string of length $\sum_{i=1}^{n}|s_i|$ formed by concatenating all strings from S. For a superstring s of S the value $\sum_{i=1}^{n}|s_i| - |s|$ is called by *compression of s with respect to S*. Then finding a shortest superstring is equivalent to finding an order of $s_1, \dots, s_n$ such that the consecutive strings have the largest possible total overlap. We first show that it is FPT with respect to r to check whether one can achieve a compression at least r by construction a kernel of size $O(r^4)$. We complement this result by a hardness result about "stronger" parameterization. Let us partition n input strings into $n/2$ pairs such that the sum of the $n/2$ resulting overlaps is maximized. Such a partition can be found in polynomial time by constructing a maximum weight matching in an auxiliary graph. Then this total overlap provides a lower bound on the maximum compression (or, equivalently, an upper bound on the length of a shortest superstring). We show that already deciding whether at least one additional symbol can be saved beyond the maximum weight matching value is already NP-complete.

2 Basic Definitions and Preliminaries

Strings. Let s be a string. By $|s|$ we denote the *length* of s. By $s[i]$, where $1 \le i \le |s|$, we denote the i-th symbol of s, and $s[i,j] = s[i] \dots s[j]$ for $1 \le i \le j \le |s|$. We assume that $s[i,j]$ is the empty string if $i > j$. We denote $\text{prefix}_i(s) = s[1,i]$ and $\text{suffix}_i(s) = s[|s|-i+1, |s|]$ the *i-th prefix* and *i-th suffix* of s respectively for $i \in \{1, \dots, |s|\}$; $\text{prefix}_0(s) = \text{suffix}_0(s)$ is the empty string. Let s, s' be strings. We write $s \subseteq s'$ to denote that s is a *substring* of s'. If $s \subseteq s'$, then s' is a *superstring* of s. We write $s \subset s'$ and $s \supset s'$ to denote proper sub and superstrings. For a collection of strings S, a string s is a superstring of S if s is a superstring of each string in S. The *compression measure* of a superstring s of a collection of strings S is $\sum_{x \in S} |x| - |s|$. If $s \subseteq s'$, then $\text{overlap}(s, s') = \text{overlap}(s', s) = s$; otherwise, if $s \not\subseteq s'$ and $s' \not\subseteq s$, then $\text{overlap}(s, s') = \text{suffix}_r(s) = \text{prefix}_r(s')$, where $r = \max\{i \mid 0 \le i \le \min\{|s|, |s'|\}, \text{suffix}_i(s) = \text{prefix}_i(s')\}$. We denote by ss' the *concatenation* of s and s'. For strings s, s', we define the *concatenation with overlap* $s \circ s'$ as follows. If $s \subseteq s'$, then $s \circ s' = s' \circ s = s'$. If $s \not\subseteq s'$ and $s' \not\subseteq s$, then $s \circ s' = \text{prefix}_p(s)\text{overlap}(s, s')\text{suffix}_q(s')$, where $p = |s| - |\text{overlap}(s, s')|$ and $q = |s'| - |\text{overlap}(s, s')|$.

We need the following folklore property of superstrings.

Lemma 1. *Let s be a superstring of a collection S of strings. Let $S' = \{s_1, \dots, s_n\}$ be a set of inclusion maximal pairwise distinct strings of S such that each string of S is a substring of a string from S'. Let indices $p_i, q_i \in \{1, \dots, |s|\}$ be such that $s_i = s[p_i, q_i]$ for $i \in \{1, \dots, n\}$ and assume that $p_1 < \dots < p_n$. Then $s' = s_1 \circ \dots \circ s_n$ is a superstring of S of length at most $|s|$.*

Graphs. We consider finite directed and undirected graphs without loops or multiple edges. The vertex set of a (directed) graph G is denoted by $V(G)$, the edge set of an undirected graph and the arc set of a directed graph G is denoted by $E(G)$. To distinguish edges and arcs, the edge with two end-vertices u, v is denoted by $\{u, v\}$, and we write (u, v) for the corresponding arc. For an arc $e = (u, v)$, v is the *head* of e and u is the tail. Let G be a directed graph. For a vertex $v \in V(G)$, we say that u is an *in-neighbor* of v if $(u, v) \in E(G)$. The set of all in-neighbors of v is denoted by $N_G^-(v)$. The *in-degree* $d_G^-(v) = |N_G^-(v)|$. Respectively, u is an *out-neighbor* of v if $(v, u) \in E(G)$, the set of all out-neighbors of v is denoted by $N^+_G(v)$, and the *out-degree* $d^+_G(v) = |N^+_G(v)|$. For a directed graph G, a (directed) *trail* of *length* k is a sequence $v_0, e_1, v_1, e_2, \ldots, e_k, v_k$ of vertices and arcs of G such that $v_0, \ldots, v_k \in V(G)$, $e_1, \ldots, e_k \in E(G)$, the arcs $e_1, \ldots, e_k$ are pairwise distinct, and for $i \in \{1, \ldots, k\}$, $e_i = (v_{i-1}, v_i)$. We omit the word "directed" if it does not create a confusion. Slightly abusing notations we often write a trail as a sequence of its vertices $v_0, \ldots, v_k$ or arcs $e_1, \ldots, e_k$. If $v_0, \ldots, v_k$ are pairwise distinct, then $v_0, \ldots, v_k$ is a (directed) path. Recall that a path of length $|V(G)| - 1$ is a *Hamiltonian* path. For an undirected graph G, a set $U \subseteq V(G)$ is a *vertex cover* of G if for any edge $\{u, v\}$ of G, $u \in U$ or $v \in U$. A set of edges M with pairwise distinct end-vertices is a *matching*.

Parameterized Complexity. Parameterized complexity is a two dimensional framework for studying the computational complexity of a problem. One dimension is the input size and another one is a parameter. We refer to the books of Downey and Fellows [5], Flum and Grohe [8], and Niedermeier [19] for detailed introductions to parameterized complexity.

Formally, a parameterized problem $\mathcal{P} \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet, i.e., an instance of $\mathcal{P}$ is a pair (I, k) for $I \in \Sigma^*$ and $k \in \mathbb{N}$, where I is an input and k is a parameter. It is said that a problem is *fixed parameter tractable* (or FPT), if it can be solved in time $f(k) \cdot |I|^{O(1)}$ for some function f. A *kernelization* for a parameterized problem is a polynomial algorithm that maps each instance (I, k) to an instance (I', k') such that

(i) (I, k) is a yes-instance if and only if (I', k') is a yes-instance of the problem, and
(ii) the size of I' and k' are bounded by $f(k)$ for a computable function f.

The output (I', k') is called a *kernel*. The function f is said to be a *size* of a kernel. Respectively, a kernel is *polynomial* if f is polynomial. While a parameterized problem is FPT if and only if it has a kernel, it is widely believed that not all FPT problems have polynomial kernels.

We use randomized algorithms for our problems. Recall that a *Monte Carlo algorithm* is a randomized algorithm whose running time is deterministic, but whose output may be incorrect with a certain (typically small) probability. A Monte-Carlo algorithm is *true-biased* (*false-biased* respectively) if it always returns a correct answer when it returns a yes-answer (a no-answer respectively).

3 FPT-Algorithms for Partial Superstring

In this section we show that Partial Superstring is FPT, when parameterized by k or ℓ. For technical reasons, we consider the following variant of the problem with weights:

Partial Weighted Superstring
Input: A collection of strings S over an alphabet Σ with a weight function $w \colon S \to \mathbb{N}_0$, and non-negative integers k, ℓ and W.
Question: Is there a string s of length at most ℓ such that s is a superstring of a collection of k strings $S' \subseteq S$ with $w(S') \geq W$?

Clearly, if $w \equiv 1$ and $W = k$, then we have the Partial Superstring problem.

Theorem 1. Partial Weighted Superstring *can be solved in time* $O((2e)^k \cdot kn^2 m \log W)$ *by a true-biased Monte-Carlo algorithm and in time* $(2e)^k k^{O(\log k)} \cdot n^2 \log n \cdot m \log W$ *by a deterministic algorithm for a collection of* n *strings of length at most* m.

Proof. First, we describe the randomized algorithm and then explain how it can be derandomized. The algorithm uses the color coding technique proposed by Alon, Yuster and Zwick [2].

If $\ell \geq km$, then the problem is trivial, as the concatenation of any k strings of S has length at most ℓ and we can greedily choose k strings of maximum weight. Assume that $\ell < km$.

We color the strings of S by k colors $1, \ldots, k$ uniformly at random independently from each other. Now we are looking for a string s that is a superstring of k strings of maximum total weight that have pairwise distinct colors.

To do it, we apply the dynamic programming across subsets. For simplicity, we explain only how to solve the decision problem, but our algorithm can be modified to find a colorful superstring as well. For $X \subseteq \{1, \ldots, k\}$, a string $x \in S$ and a positive integer $h \in \{1, \ldots, \ell\}$, the algorithm computes the maximum weight $W(X, x, h)$ of a string s of length at most h such that

(i) s is a superstring of a collection of $k' = |X|$ strings $S' \subseteq S$ of pairwise distinct colors from X,
(ii) x is inclusion maximal string of S' and $x = \text{suffix}_{|x|}(s)$.

If such a string s does not exist, then $W(X, x, h) = -\infty$.

We compute the table of values of $W(X, x, h)$ consecutively for $|X| = 1, \ldots, k$. To simplify computations, we assume that $W(X, x, h) = -\infty$ for $h < 0$. If $|X| = 1$, then for each string $x \in S$, we set $W(X, x, h) = w(x)$ if x is colored by the unique color of X and $|x| \leq h$. In all other cases $W(X, x, h) = -\infty$. Assume that $|X| = k' \geq 2$ and the values of $W(X', x, h)$ are already computed for $|X'| < k'$. Let

$$W' = \max\{W(X \setminus \{c\}, x, h) + w(y) \mid y \subseteq x \text{ has color } c \in X\},$$

and

$$W'' = \max\{W(X \setminus \{c\}, y, h - |x| + |\text{overlap}(y, x)|) + w(x) \mid x \nsubseteq y, y \nsubseteq x\},$$

where c is the color of x; we assume that $W' = -\infty$ if there is no substring y of x of color $c \in X$, and $W'' = -\infty$ if every string y is a sub or superstring of x. We set $W(X, x, h) = \max\{W', W''\}$.

We show that $\max\{W(\{1, \ldots, k\}, x, \ell) \mid x \in S\}$ is the maximum weight of k strings of S colored by distinct colors that have a superstring of length at most ℓ; if this value equals $-\infty$, then there is no string of length at most ℓ that is a superstring of k string of S of distinct colors.

To prove this, it is sufficient to show that the values $W(X, x, h)$ computed by the algorithms are the maximum weights of strings of length at most h that satisfy (i) and (ii). The proof is by induction on the size of $|X|$. It is straightforward to verify that it holds if $|X| = 1$. Assume that $|X| > 1$ and the claim holds for sets of lesser size. Denote by $W^*(X, x, h)$ the maximum weight of a string s of length at most h that satisfies (i) and (ii). By the description of the algorithm, $W^*(X, x, h) \geq W(X, x, h)$. We show that $W^*(X, x, h) \leq W(X, x, h)$.

Let S' be a collection of k' strings of pairwise distinct colors from X that have s as a superstring. Denote by S'' a set of inclusion maximal distinct strings of S' that contains x such that every string of S' is a substring of a string of S''. Assume that $S'' = \{x_1, \ldots, x_r\}$ and $x_i = s[p_i, q_i]$ for $i \in \{1, \ldots, r\}$. Clearly, $x = x_r$.

Suppose that there is $y \in S' \setminus \{x\}$ such that $y \subseteq x$. Let $c \in X$ be a color of y. Then s is a superstring of $S' \setminus \{y\}$ and the total weight of these string is $W^*(X, x, h) - w(y)$. By induction, $W^*(X, x, h) - w(y) \leq W(X \setminus \{c\}, x, h)$ and we have that $W^*(X, x, h) \leq W(X \setminus \{c\}, x, h) + w(y) \leq W' \leq W(X, x, h)$.

Suppose now that $S' \setminus \{x\}$ does not contain substrings of x. Then $r \geq 2$. Let $y = s_{r-1}$ and $s' = s[1, q_{i-1}]$. Observe that $y = \text{suffix}_{|y|}(s')$. Notice that s' is a superstring of $S'' \setminus x$. Because $S' \setminus \{x\}$ has no substrings of x, every string in $S' \setminus \{x\}$ is a substring of any superstring of $S'' \setminus \{x\}$ and, therefore, s' is a superstring of $S' \setminus \{x\}$ of length at most $|s| - |x| + |\text{overlap}(y, x)| \leq h - |x| + |\text{overlap}(y, x)|$. The weight of $S' \setminus \{x\}$ is $W^*(X, x, h) - w(x)$. By induction, $W^*(X, x, h) - w(x) \leq W(X \setminus \{c\}, y, h - |x| + |\text{overlap}(y, x)|)$. Hence $W^*(X, x, h) \leq W(X \setminus \{c\}, y, h - |x| + |\text{overlap}(y, x)|) + w(x) \leq W'' \leq W(X, x, h)$.

To evaluate the running time of the dynamic programming algorithm, observe that we can check whether y is a substring of x or find overlap(y, x) in time $O(m)$ using, e.g., the algorithm of Knuth, Morris, and Pratt [15], and we can construct the table of the overlaps and their sizes in time $O(n^2 m)$. Hence, for each X, the values $W(X, x, h)$ can be computed in time $O(n^2 km \log W)$, as $h \leq \ell < km$. Therefore, the running time is $O(2^k \cdot n^2 km \log W)$.

We proved that an optimal colorful solution can be found in time $O(2^k \cdot n^2 km \log W)$. Using the standard color coding arguments (see [2]), we obtain that it is sufficient to consider $N = e^k$ random colorings of S to claim that with probability $\alpha > 0$, where α is a constant that does not depend on the input size and the parameter, we get a coloring for which k string of S that have a superstring of length at most ℓ and the total weight at least W are colored by distinct colors if such a string exists. It implies that Partial Weighted Superstring can be solved in time $O((2e)^k \cdot kn^2 m \log W)$ by our randomized algorithm.

To derandomize the algorithm, we apply the technique proposed by Alon, Yuster and Zwick [2] using the k-perfect hash functions constructed by Naor, Schulman and Srinivasan [18]. The random colorings are replaced by the family of at most $e^k k^{\log k} \log n$ hash functions $c \colon S \to \{1, \ldots, k\}$ that have the following property: there is a hash function c that colors k string of S that have a superstring of length at most ℓ and the total weight at least W by distinct colors if such a string exists. It implies that PARTIAL WEIGHTED SUPERSTRING can be solved in time $(2e)^k k^{O(\log k)} \cdot n^2 \log n \cdot m \log W$ deterministically. □

Because PARTIAL SUPERSTRING is a special case of PARTIAL WEIGHTED SUPERSTRING, Theorem 1 implies that this problem is FPT when parameterized by k. We show that the same holds if we parameterize the problem by ℓ.

Corollary 1. PARTIAL SUPERSTRING *is* FPT *when parameterized by ℓ.*

Proof. Consider an instance (S, k, ℓ) of PARTIAL SUPERSTRING. Recall that S can contain several copies of the same string. We construct a set of weighted strings S' by replacing a string s that occurs r times in S by the single copy of s of weight $w(s) = r$. Let $W = k$. Observe that there is a string s of length at most ℓ such that s is a superstring of a collection of at least k strings of S if and only if there a string s of length at most ℓ such that s is a superstring of a set of strings of S' of total weight at least W. A string of length at most ℓ has at most $\ell(\ell-1)/2$ distinct substrings. We consider the instances (S', w, k', ℓ, W) of PARTIAL WEIGHTED SUPERSTRING for $k' \in \{1, \ldots, \ell(\ell-1)/2\}$. For each of these instances, we solve the problem using Theorem 1. It remains to observe that there is a string s of length at most ℓ such that s is a superstring of a set of strings of S' of total weight at least W if and only if one of the instances (S', w, k', ℓ, W) is a yes-instance of PARTIAL WEIGHTED SUPERSTRING. □

We complement the above algorithmic results by showing that we hardly can expect that PARTIAL SUPERSTRING has a polynomial kernel when parameterized by k or ℓ.

Theorem 2. PARTIAL SUPERSTRING *does not admit a polynomial kernel when parameterized by $k+m$ or $\ell+m$ for strings of length at most m over the alphabet $\Sigma = \{0, 1\}$ unless* NP $\subseteq$ coNP /poly.

4 Shortest Superstring Below Guaranteed Values

In this section we discuss SHORTEST SUPERSTRING parameterized by the difference between upper bounds for the length of a shortest superstring and the length of a solution superstring. For a collection of strings S, the length of the shortest superstring is trivially upper bounded by $\sum_{x \in S} |x|$. We show that SHORTEST SUPERSTRING admits a polynomial kernel when parameterized by the compression measure of a solution.

Theorem 3. SHORTEST SUPERSTRING *admits a kernel of size $O(r^4)$ when parameterized by $r = \sum_{x \in S} |x| - \ell$.*

Proof. Let (S, ℓ) be an instance of SHORTEST SUPERSTRING, $r = \sum_{x \in S} |x| - \ell$. First, we apply the following reduction rules for the instance.

Rule 1. If there are distinct elements x and y of S such that $x \subseteq y$, then delete x and set $r = r - |x|$. If $r \leq 0$, then return a yes-answer and stop.

Rule 2. If there is $x \in S$ such that for any $y \in S \setminus \{x\}$, $|\text{overlap}(x, y)| = |\text{overlap}(y, x)| = 0$, then delete x and set $\ell = \ell - |x|$. If $S = \emptyset$ and $\ell \geq 0$, then return a yes-answer and stop. If $\ell < 0$, then return a no-answer and stop.

Rule 3. If there are distinct elements x and y of S such that $|\text{overlap}(x, y)| \geq r$, then return a yes-answer and stop.

It is straightforward to verify that these rules are *safe*, i.e., by the application of a rule we either solve the problem or obtain an equivalent instance. We exhaustively apply Rules 1–3. To simplify notations, we assume that S is the obtained set of strings and ℓ and r are the obtained values of the parameters. Notice that all strings in S are distinct and no string is a substring of another. Our next aim is to bound the lengths of considered strings.

Rule 4. If there is $x \in S$ with $|x| > 2r$, then set $\ell = \ell - |x| + 2r$ and $x = \text{prefix}_r(x)\text{suffix}_r(x)$. If $\ell < 0$, then return a no-answer and stop.

To see that the rule is safe, recall that x is not a sub or superstring of any other string of S, and $|\text{overlap}(x, y)| < r$ and $|\text{overlap}(y, x)| < r$ for any $y \in S$ distinct from x after the applications of Rule 3. As before, we apply Rule 4 exhaustively.

Now we construct an auxiliary graph G with the vertex set S such that two distinct $x, y \in S$ are adjacent in G if and only if $|\text{overlap}(x, y)| > 0$ or $|\text{overlap}(y, x)| > 0$. We greedily select a maximal matching M in G and apply the following rule.

Rule 5. If $|M| \geq r$, then return a yes-answer and stop.

To show that the rule is safe, it is sufficient to observe that if $M = \{x_1, x'_1\}, \ldots, \{x_h, x'_h\}$, $|\text{overlap}(x_i, x'_i)| > 0$ for $i \in \{1, \ldots, h\}$ and $h \geq r$, then the string s obtained by the consecutive concatenations with overlaps of $x_1, x'_1, \ldots, x_h, x'_h$ and then all the other strings of S in arbitrary order, then the compression measure of s is at least r.

Assume from now that we do not stop here, i.e., $|M| \leq r - 1$. Let $X \subseteq S$ be the set of end-vertices of the edges of M and $Y = S \setminus X$. Let $X = \{x_1, \ldots, x_h\}$. Clearly, $h \leq 2(r - 1)$. Observe that X is a vertex cover of G and Y is an independent set of G.

For each ordered pair (i, j) of distinct $i, j \in \{1, \ldots, h\}$, find an ordering $y_1, \ldots, y_t$ of the elements of Y sorted by the decrease of $|\text{overlap}(x_i, y_p)| + |\text{overlap}(y_p, x_j)|$ for $p \in \{1, \ldots, t\}$. We construct the set $R_{(i,j)}$ that contains the first $\min\{2h, t\}$ elements of the sequence.

For each $i \in \{1, \ldots, h\}$, find an ordering $y_1, \ldots, y_t$ of the elements of Y sorted by the decrease of $|\text{overlap}(y_p, x_i)|$ for $p \in \{1, \ldots, t\}$. We construct the set S_i that contains the first $\min\{2h, t\}$ elements of the sequence.

For each $i \in \{1, \ldots, h\}$, find an ordering $y_1, \ldots, y_t$ of the elements of Y sorted by the decrease of $|\text{overlap}(x_i, y_p)|$ for $p \in \{1, \ldots, t\}$. We construct the set T_i that contains the first $\min\{2h, t\}$ elements of the sequence.

Let

$$S' = X \cup \Big(\bigcup_{(i,j),\ i,j \in \{1,\ldots,h\}, i \neq j} R_{(i,j)} \Big) \cup \Big(\bigcup_{i \in \{1,\ldots,h\}} S_i \Big) \cup \Big(\bigcup_{i \in \{1,\ldots,h\}} T_i \Big).$$

Claim $(*)$**.** *There is a superstring s of S with the compression measure at least r if and only if there is a superstring s' of S' with the compression measure at least r.*

Proof (of Claim $(*)$). If s' is a superstring of S' with the compression measure at least r, then the string s obtained from s' by the concatenation of s' and the strings of $S \setminus S'$ (in any order) is a superstring of S with the same compression measure as s'.

Suppose that s is a shortest superstring of S and the compression measure at least r. By Lemma 1, $s = s_1 \circ \ldots \circ s_n$, where $S = \{s_1, \ldots, s_n\}$. Let

$$Z = \{s_i \mid s_i \in Y, |\text{overlap}(s_{i-1}, s_i)| > 0 \text{ or } |\text{overlap}(s_i, s_{i+1})| > 0, 1 \leq i \leq n\};$$

we assume that s_0, s_{n+1} are empty strings.

We show that $|Z| \leq 2h$. Suppose that $s_i \in Z$. If $|\text{overlap}(s_{i-1}, s_i)| > 0$, then $s_{i-1} \in X$, because $s_i \in Y$ and any two strings of Y have the empty overlap. By the same arguments, if $|\text{overlap}(s_i, s_{i+1})| > 0$, then $s_{i+1} \in X$. Because $|X| = h$, we have that $|Z| \leq 2h$.

Suppose that the shortest superstring s is chosen in such a way that $|Z \setminus S'|$ is minimum. We prove that $Z \subseteq S'$ in this case. To obtain a contradiction, assume that there is $s_i \in Z \setminus S'$. We consider three cases.

Case 1. $|\text{overlap}(s_{i-1}, s_i)| > 0$ and $|\text{overlap}(s_i, s_{i+1})| > 0$. Recall that $s_{i-1}, s_{i+1} \in X$ in this case. Since $s_i \notin S'$, $s_i \notin R_{(p,q)}$ for $x_p = s_{i-1}$ and $x_q = s_{i+1}$. In particular, it means that $|R_{(p,q)}| = 2h$. As $|Z| \leq 2h$ and $|R_{(p,q)}| = 2h$, there is $s_j \in R_{(p,q)}$ such that $s_j \notin Z$, i.e., $|\text{overlap}(s_{j-1}, s_j)| = |\text{overlap}(s_j, s_{j+1})| = 0$. By the definition of $R_{(p,q)}$, $|\text{overlap}(s_{i-1}, s_j)| + |\text{overlap}(s_j, s_{i+1})| \geq |\text{overlap}(s_{i-1}, s_i)| + |\text{overlap}(s_i, s_{i+1})|$. Consider $s^* = s_1 \circ \ldots \circ s_{i-1} \circ s_j \circ s_{i+1} \ldots \circ s_{j-1} \circ s_i \circ s_j \circ \ldots \circ s_n$ assuming that $i < j$ (the other case is similar). Because $|\text{overlap}(s_{i-1}, s_j)| + |\text{overlap}(s_j, s_{i+1})| \geq |\text{overlap}(s_{i-1}, s_i)| + |\text{overlap}(s_i, s_{i+1})|$, $|s^*| \leq |s|$. Moreover, since s is a shortest superstring of S, $|s| \geq |s^*|$ and, therefore, $|\text{overlap}(s_{j-1}, s_i)| = |\text{overlap}(s_i, s_{j+1})| = 0$. But then for the set Z^* constructed for s^* in the same way as the set Z for s, we obtain that $|Z^* \setminus S'| < |Z \setminus S'|$; a contradiction.

Case 2. $|\text{overlap}(s_{i-1}, s_i)| = 0$ and $|\text{overlap}(s_i, s_{i+1})| > 0$. Then $s_{i+1} \in X$. Since $s_i \notin S'$, $s_i \notin S_p$ for $x_p = s_{i+1}$ and $|S_p| = 2h$. As $|Z| \leq 2h$ and $|S_p| = 2h$, there is $s_j \in S_p$ such that $s_j \notin Z$, i.e., $|\text{overlap}(s_{j-1}, s_j)| = |\text{overlap}(s_j, s_{j+1})| = 0$. By the definition of S_p, $|\text{overlap}(s_j, s_{i+1})| \geq |\text{overlap}(s_i, s_{i+1})|$. As in Case 1, consider s^* obtained by the exchange of s_i and s_j in the sequence of strings that is used for the concatenations with overlaps. In the same way, we obtain a contradiction with the choice of Z, because for the set Z^* constructed for s^* in the same way as the set Z for s, we obtain that $|Z^* \setminus S'| < |Z \setminus S'|$.

Case 3. $|\text{overlap}(s_{i-1}, s_i)| > 0$ and $|\text{overlap}(s_i, s_{i+1})| = 0$. To obtain contradiction in this case, we use the same arguments as in Case 2 using symmetry. Notice that we should consider T_p instead of S_p.

Now let $s' = s_{i_1} \circ \ldots \circ s_{i_p}$, where $s_{i_1}, \ldots, s_{i_p}$ is the sequence of string of S' obtained from $s_1, \ldots, s_n$ by the deletion of the strings of $S \setminus S'$. Because we have

that $Z \subseteq S'$, the overlap of each deleted string with its neighbors is empty and, therefore, s' has the same compression measure as s
To finish the construction of the kernel, we define $\ell' = \ell - \sum_{x \in S \setminus S'} |x|$ and apply the following rule that is safe by Claim $(*)$.

Rule 6. If $\ell' < 0$, then return a no-answer and stop. Otherwise, return the instance (S', ℓ') and stop.

Since $|X| = h \leq 2(r-1)$, $|S'| \leq h + h^2 \cdot 2h + h \cdot 2h + h \cdot 2h = 2h^3 + 4h^2 + h = O(h^3) = O(r^3)$. Because each string of S' has length at most $2r$, the kernel has size $O(r^4)$.

It is easy to see that Rules 1-3 can be applied in polynomial time. Then graph G and M can be constructed in polynomial time and, trivially, Rule 5 demands $O(1)$ time. The sets X, Y, $R_{(i,j)}$, S_i and T_i can be constructed in polynomial time. Hence, S' and ℓ' can be constructed in polynomial time. Because Rule 6 can be applied in time $O(1)$, we conclude that the kernel is constructed in polynomial time. □

Now we consider another upper bound for the length of the shortest superstring. Let S be a collection of strings. We construct an auxiliary weighted graph $G(S)$ with the vertex set S by assigning the weight $w(\{x, y\}) = \max\{|\text{overlap}(x, y)|, |\text{overlap}(y, x)|\}$ for any two distinct $x, y \in S$. Let $\mu(S)$ be the size of a maximum weighted matching in G. Clearly, G can be constructed in polynomial time and the computation of $\mu(G)$ is well known to be polynomial [6]. If $M = \{x_1, y_1\}, \ldots, \{x_h, y_h\}$ and $|\text{overlap}(x_i, y_i)| = w(\{x_i, y_i\})$ for $i \in \{1, \ldots, h\}$, then the string s obtained by the consecutive concatenations with overlaps of $x_1, y_1, \ldots, x_h, y_h$ and then (possibly) the remaining string of S has the compression measure at least $\mu(G)$. Hence, $\sum_{x \in S} |x| - \mu(G)$ is the upper bound for the length of the shortest superstring of G. We show that it is NP-hard to find a superstring that is shorter than this bound.

Theorem 4. Shortest Superstring *is* NP-*complete for* $\ell = \sum_{x \in S} |x| - \mu(S) - 1$ *even if restricted to the alphabet* $\Sigma = \{0, 1\}$.

References

1. Concorde TSP Solver. http://www.math.uwaterloo.ca/tsp/concorde.html
2. Alon, N., Yuster, R., Zwick, U.: Color-coding. J. ACM **42**(4), 844–856 (1995)
3. Bellman, R.: Dynamic programming treatment of the travelling salesman problem. J. ACM (JACM) **9**(1), 61–63 (1962)
4. Bulteau, L., Hüffner, F., Komusiewicz, C., Niedermeier, R.: Multivariate algorithmics for NP-hard string problems. Bull. EATCS **114**, 31–73 (2014)
5. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Texts in Computer Science. Springer, Berlin (2013)
6. Edmonds, J.: Maximum matching and a polyhedron with 0, 1-vertices. J. Res. Nat. Bur. Standards Sect. B **69B**, 125–130 (1965)
7. Evans, P.A., Wareham, T.: Efficient restricted-case algorithms for problems in computational biology. In: Zomaya, A.Y., Elloumi, M. (eds.) Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications. Wiley Series in Bioinformatics, pp. 27–49. Wiley, Chichester (2011)

8. Flum, J., Grohe, M.: Parameterized Complexity Theory. Texts in Theoretical Computer Science. An EATCS Series. Springer-Verlag, Berlin (2006)
9. Gallant, J., Maier, D., Storer, J.A.: On finding minimal length superstrings. J. Comput. Syst. Sci. **20**(1), 50–58 (1980)
10. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, New York (1979)
11. Golovnev, A., Kulikov, A.S., Mihajlin, I.: Solving SCS for bounded length strings in fewer than 2^n steps. Inf. Process. Lett. **114**(8), 421–425 (2014)
12. Golovnev, A., Kulikov, A.S., Mihajlin, I.: Solving 3-superstring in $3^{\mathrm{n}/3}$ time. In: Chatterjee, K., Sgall, J. (eds.) MFCS 2013. LNCS, vol. 8087, pp. 480–491. Springer, Heidelberg (2013)
13. Held, M., Karp, R.M.: A dynamic programming approach to sequencing problems. J. Soc. Ind. Applied Math. **10**(1), 196–210 (1962)
14. Karp, R.M.: Dynamic programming meets the principle of inclusion and exclusion. Oper. Res. Lett **1**(2), 49–51 (1982)
15. Knuth, D.E., Morris, J.H.J., Pratt, V.R.: Fast pattern matching in strings. SIAM J. Comput. **6**(2), 323–350 (1977)
16. Kohn, S., Gottlieb, A., Kohn, M.: A generating function approach to the traveling salesman problem. In: Proceedings of the 1977 Annual Conference, pp. 294–300. ACM (1977)
17. Mucha, M.: Lyndon words and short superstrings. In: Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 958–972. SIAM (2013)
18. Naor, M., Schulman, L.J., Srinivasan, A.: Splitters and near-optimal derandomization. In: FOCS, pp. 182–191. IEEE Computer Society (1995)
19. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Mathematics and its Applications, vol. 31. Oxford University Press, Oxford (2006)