A $2\frac{2}{3}$ Superstring Approximation Algorithm

Chris Armen*
University of Hartford
W. Hartford, CT 06117-1599

Clifford Stein†
Dartmouth College
Hanover, NH 03755-3510

March 17, 1998

Abstract

Given a collection of strings $S = \{s_1, \ldots, s_n\}$ over an alphabet Σ, a *superstring* α of S is a string containing each s_i as a substring; that is, for each i, $1 \leq i \leq n$, α contains a block of $|s_i|$ consecutive characters that match s_i exactly. The *shortest superstring problem* is the problem of finding a superstring α of minimum length.

The shortest superstring problem has applications in both data compression and computational biology. It was shown by Blum et al. [3] to be MAX SNP-hard. The first $O(1)$-approximation algorithm also appeared in [3], which returns a superstring no more than 3 times the length of an optimal solution. Prior to the algorithm described in this paper, there were several published results that improved on the approximation ratio; of these, the best was our algorithm ShortString, a $2\frac{3}{4}$–approximation [1].

We present our new algorithm, G-ShortString, which achieves an approximation ratio of $2\frac{2}{3}$. Our approach builds on the work in [1], in which we identified classes of strings that have a nested periodic structure, and which must be present in the worst case for our algorithms. We introduced machinery to describe these strings and proved strong structural properties about them. In this paper we extend this study to strings that exhibit a more relaxed form of the same structure, and we use this understanding to obtain our improved result.

1 Introduction

Given a collection of strings $S = \{s_1, \ldots, s_n\}$ over an alphabet Σ, a *superstring* α of S is a string containing each s_i as a substring, that is, for each i, $1 \leq i \leq n$, α contains $|s_i|$ consecutive characters that match s_i exactly. The *shortest superstring problem* is the problem of finding a superstring α of minimum length.

The shortest superstring problem has applications in both computational biology [7, 16, 18] and data compression [10, 20]. We begin by briefly describing the former. DNA sequencing is the task of determining the sequence of nucleotides in a molecule of DNA. These nucleotides are one of adenine, cytosine, guanine, and thymine, and are typically represented by the alphabet $\{a, c, g, t\}$. A molecule of human DNA is about 10^8 nucleotides long. Current laboratory procedures can directly determine the nucleotides of a fragment of DNA up to about 600 nucleotides long. In *shotgun sequencing*, several copies of a DNA molecule are fragmented using techniques such as sonication or exposure to various restriction enzymes.

*Contact author. Email: armen@hartford.edu.

†Email: cliff@cs.dartmouth.edu. Research partly supported by NSF Award CCR-9308701, a Walter Burke Research Initiation Award and a Dartmouth College Research Initiation Award.

Once the nucleotides of all of the fragments have been determined, the *sequence assembly problem* is the computational task of reconstructing the original molecule from the overlapping fragments. The shortest superstring problem is an abstraction of this problem, in which the shortest reconstruction is assumed to be the most likely on the grounds that it is the most parsimonious.

The shortest superstring problem was shown to be NP-hard by Gallant, Maier and Storer [10]; it was later shown to be MAX SNP-hard [3]. The first $O(1)$-approximations were given by Blum et al. in [3], who showed that a greedy algorithm always returns a string that is no longer than four times optimal; they also give a modified greedy algorithm which returns a string that is no more three times the optimal length. Other algorithms were later shown to produce approximations of $2\frac{8}{9}$, $2\frac{5}{6}$, and slightly better than 2.8 (Teng and Yao [22], Czumaj et al. [6], and Kosaraju et al. [15], respectively). Our result of $2\frac{3}{4}$ [1, 2] was the best known until recently, and when combined with the algorithm of [15] achieves an approximation ratio of about 2.725.

In this paper we describe our $2\frac{2}{3}$-approximation algorithm for the shortest superstring problem. Algorithmically, our approach is a generalization of the one taken in [1], but the analysis is very different. Our approach is interesting in its own right as an original contribution to string combinatorics. We introduce techniques for the analysis of complex periodic strings that may be of use in attaining subsequent improvements to the approximation ratio.

We now give a brief overview of our approach. All of the above mentioned algorithms begin by finding a minimum-weight cycle cover on a graph which has a node for every string and an edge between string u and v of length $|u| - \mathrm{ov}(u, v)$, where $\mathrm{ov}(u, v)$ is the amount of overlap that can be obtained by merging u and v. This cycle cover partitions the strings into cycles; the remaining work is in patching the cycles together to form one cycle covering the whole graph. The key to our new algorithm is to exploit the periodic structure of the cycles of strings that arise in this problem. In particular, the 3-approximation of [3] uses a theorem about infinite periodic functions [8], and the correspondence between periodic functions and strings in cycles. However, the particular instances of cycle patching that appear to be difficult actually involve short periodic strings, that is, strings that are periodic, but whose period may repeat only slightly more than once. We prove several new properties about such strings, allowing us to answer questions of the following form: given a string with some periodic structure, characterize *all* the possible periodic strings that can have a large amount of overlap with the first string. Given this understanding, we will be able to predict the ways in which overlap between certain strings can occur, and thus plan for it algorithmically.

There have been two subsequent results on this problem. Independently of our work, Breslauer et al.[4] obtained a 2.596-approximation by proving other structural characteristics of periodic strings. More recently, Z Sweedyk [21] has obtained a $2\frac{1}{2}$ approximation.

Beyond the goal of devising improved approximation algorithms, there are important open questions that remain. One of these pertains to the greedy algorithm that is often used in practice. This algorithm repeatedly merges a pair of strings with the maximum amount of overlap. No one has produced an example showing that this algorithm produces a superstring more than twice as long as an optimal solution, yet the strongest known result [3] is that it is a 4-approximation. Thus it is often conjectured that the greedy algorithm is a 2-approximation; this remains open. Another open question is the existence of a $(2 - \epsilon)$-approximation, or a lower bound result.

2 Preliminaries

For consistency, we use some notation and definitions of [3] and [22]. We introduce this notation rather tersely; the reader interested in a more detailed introduction is referred to [3] or [**?**]. We assume, without loss of generality, that the set S of strings is *substring free*; i.e., no s_j is a substring of s_i, $i \neq j$. We use $|s_i|$ to denote the length of string s_i, $|S|$ to denote the sum of the lengths of all the strings, and $\mathrm{opt}(S)$ to denote the length of the shortest superstring of S.

Given two strings s and t, we define $\mathrm{ov}(s,t)$, the *overlap* between s and t, to be the length of the longest string x, such that there exist non-empty u and v with $s = ux$ and $t = xv$. (Recall that our set S is substring-free.) We call u the *prefix* of s with respect to t, $\mathrm{pref}(s,t)$, and refer to $|u|$ as the *prefix distance* from s to t, $\mathrm{d_{pref}}(s,t)$. Observe that for any s and t, $\mathrm{ov}(s,t) + \mathrm{d_{pref}}(s,t) = |s|$. String uxv, the shortest superstring of s and t in which s appears before t is denoted by $\langle s,t\rangle$, and $|\langle s,t\rangle| = |s| + |t| - \mathrm{ov}(s,t)$.

We can map the superstring problem to a graph problem by defining the *distance graph.* We create a graph $G = (V,E)$ with a vertex $v_i \in V$ for each string $s_i \in S$. For every ordered pair of vertices v_i, v_j, we place a directed edge of length $\mathrm{d_{pref}}(s_i,s_j)$ and label the edge with $\mathrm{pref}(s_i,s_j)$. We can now observe that a minimum length hamiltonian cycle (traveling salesman tour) $v_{\pi_1},\ldots,v_{\pi_n},v_{\pi_1}$, in G, with edge (i,j) labeled by $\mathrm{pref}(s_{\pi_i},s_{\pi_j})$, almost corresponds to a superstring in S, the only difference being that we must replace $\mathrm{pref}(s_{\pi_n},s_{\pi_1})$ with s_{π_n}. Since $|\mathrm{pref}(s_i,s_j)| \le |s_i|$, we can conclude that $\mathrm{opt}(TSP) \le \mathrm{opt}(S)$, where $\mathrm{opt}(TSP)$ is the optimal solution to TSP defined above. This TSP is directed (sometimes called *asymmetric*); thus the best known approximation [9] is only within a factor of $O(\log n)$. Therefore, we must exploit more of the structure of the problem in order to achieve better bounds.

Given a directed graph G, with weights on the edges, a *cycle cover* C is a set of cycles such that each vertex is in exactly one cycle. A minimum-cost cycle cover is a cycle cover such that the sum of the weights of the edges in all the cycles is minimized; let $\mathrm{d}(C)$ be total weight of a minimum-cost cycle cover C. A minimum-cost cycle cover can be computed in $O(n^3)$ time by a well-known reduction to the assignment problem [17]. Since a tour is a cycle cover, $\mathrm{d}(C) \le \mathrm{opt}(TSP)$. As noted above, $\mathrm{opt}(TSP) \le \mathrm{opt}(S)$, so the weight of the cycle cover $\mathrm{d}(C)$ gives us a lower bound on the length of the optimal solution $\mathrm{opt}(S)$.

Because $\mathrm{d_{pref}}(s_i,s_j) + \mathrm{ov}(s_i,s_j) = |s_i|$, one could also weight the edges by their overlap, find a maximum-cost cycle cover and obtain the same solution. A superstring that has minimum length, or distance, also has maximum overlap. However, this correspondence breaks down for approximations; approximating the largest overlap appears to be an easier problem (cf. [23, 22, 15]) than approximating the shortest superstring.

For two strings s and t, we write st to denote the concatenation of s and t; equivalently, we will sometimes write $s \cdot t$ for emphasis or typographical clarity. We use the conventional notation t^k to denote the concatentation of k copies of a string t, and t^∞ to denote the semi-infinite string $ttt\ldots$. When we say that a string s_i is in some cycle c of cycle cover C, or write $s_i \in c$, we mean that the vertex v_i with which s_i is associated is in cycle c. Throughout the paper, when we refer to a cycle, we will be referring to a cycle that is in a minimum-cost cycle cover in the distance graph.

We can view cycles as generators of strings. That is, informally, we can view the string s_i associated with vertex v_i in a cycle c as beginning at v_i in c, and "going around" c some (not necessarily integral) number of times. As s_i goes around c, the characters of s_i match the prefix labels around c. Let $\mathrm{per}(c)$ be any string formed by concatenating all of the labels on the edges of a cycle c. We note that there are many choices for $\mathrm{per}(c)$, all of which can be obtained by choosing an arbitrary one and performing rotations upon it. In context, it will be clear whether we mean a specific one of these rotations, or an arbitrary one. (This deliberate ambiguity is employed to simplify notation.) Note, then, that for each string $s \in c$, s is a substring of $\mathrm{per}(c)^\infty$.

Let $\mathrm{d}(c) = \sum_{e\in c} \mathrm{d_{pref}}(e)$ be the sum of the weights of the edges around a cycle c. Because of the correspondence between the $\mathrm{d_{pref}}(,)$ and $\mathrm{pref}(,)$ functions, we also have $|\mathrm{per}(c)| = \mathrm{d}(c)$. Recall that $\mathrm{d}(C)$ is the total weight (in the distance graph) of a cycle cover C; therefore $\mathrm{d}(C) = \sum_{c\in C} \mathrm{d}(c)$.

A cycle cover has the effect of grouping together similar strings. We will frequently be interested in transforming a cycle c into a superstring of the strings in c; this can also be viewed as breaking

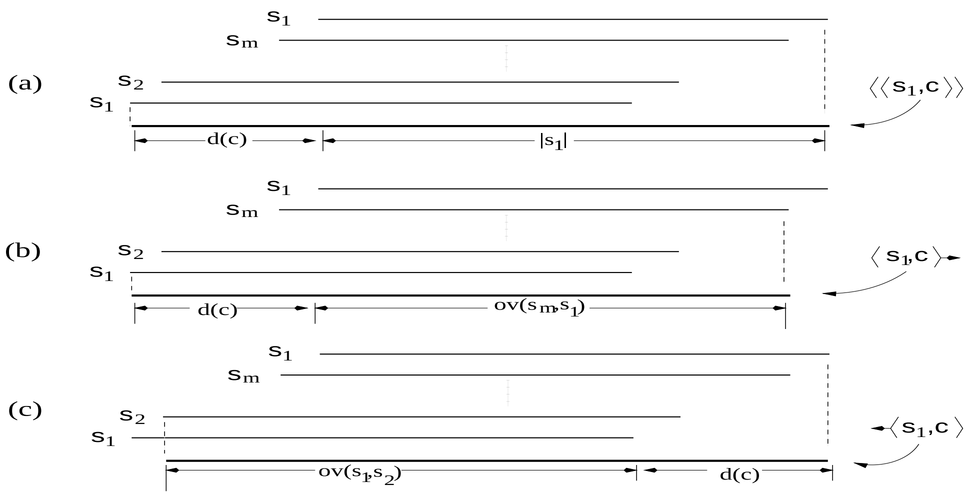


Figure 1: Types of extension. (a) Full Extension. (b) Right Extension. (c) Left Extension.

a cycle at some edge. We introduce the following notation to describe different ways of breaking a cycle and making this transformation. Let c be an m-vertex cycle containing strings $s_1, \ldots, s_m$ indexed in order around c. We define:

Full extension:

$$\langle\langle s_1, c\rangle\rangle \;=\; \langle s_1, s_2, \ldots, s_m, s_1\rangle \;=\; s_1 \cdot \mathrm{per}(c) = \mathrm{per}(c) \cdot s_1,$$

$$|\langle\langle s_1, c\rangle\rangle| = \mathrm{d}(c) + |s_1| \tag{1}$$

Right extension:

$$\langle s_1, c\rangle\!\!\rightarrow \;=\; \langle s_1, s_2, \ldots, s_m\rangle \;=\; \mathrm{pref}(s_1, s_2) \cdot \mathrm{pref}(s_2, s_3) \cdot \ldots \cdot \mathrm{pref}(s_{m-1}, s_m) \cdot s_m,$$

$$|\langle s_1, c\rangle\!\!\rightarrow| = \mathrm{d}(c) + \mathrm{ov}(s_m, s_1) \tag{2}$$

Left extension:

$$\leftarrow\!\!\langle s_1, c\rangle \;=\; \langle s_2, s_3, \ldots, s_m, s_1\rangle \;=\; \mathrm{pref}(s_2, s_3) \cdot \mathrm{pref}(s_3, s_4) \cdot \ldots \cdot \mathrm{pref}(s_m, s_1) \cdot s_1,$$

$$|\leftarrow\!\!\langle s_1, c\rangle| = \mathrm{d}(c) + \mathrm{ov}(s_1, s_2) \tag{3}$$

The three types of extension are illustrated in Figure 1 for string s_1. The definitions of $\langle\langle s_i, c\rangle\rangle$, $\langle s_i, c\rangle\!\!\rightarrow$, and $\leftarrow\!\!\langle s_i, c\rangle$, $2 \leq i \leq m$, are analogous. Notice that $\langle s_1, c\rangle\!\!\rightarrow = \leftarrow\!\!\langle s_m, c\rangle$; this apparent redundancy will be useful later. Full extension, right extension and left extension each form superstrings that include all of the strings in c, as proved in [3]. Left and right extension form shorter superstrings than full extension, though perhaps by only one character. Full extension has one additional important characteristic; $\langle\langle s_1, c\rangle\rangle$ is a superstring of the strings in c in which s_1 appears twice, once as a prefix and once as a suffix of $\langle\langle s_1, c\rangle\rangle$.

We now describe a generic version of a superstring algorithm from [3] that is also used in some form by [22], [2], and [6]. An execution of the algorithm appears as Fig. 2.

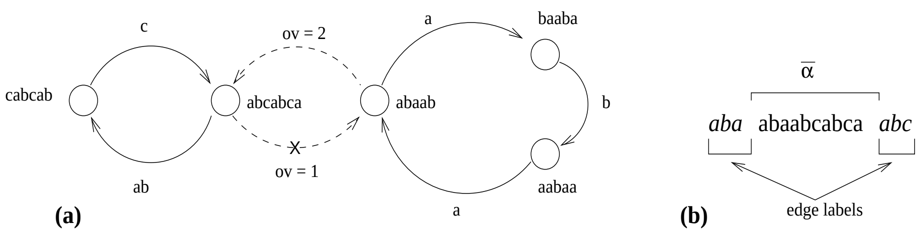


Figure 2: Execution of Generic SuperString Algorithm. Nodes are labeled with strings, edges with prefixes. (a) The graph after Step (4). Solid edges are in C, dashed edges in CC. The edge with an X is the one discarded in Step (4). (b) The final string consisting of $\bar{\alpha}$ (the merge of *abaab* and *abcabca*) along with the labels from the edges of the cycles.

<u>Generic SuperString Algorithm</u>

(1) Find a minimum cost cycle cover C in the distance graph G.
(2) For each cycle $c \in C$, choose one string to be a representative r_c.
  Let G' be the subgraph induced by the representative set R.
(3) Compute a cycle cover CC on G'.
(4) For each cycle $\gamma_i \in CC$
  Let (w_i, x_i) be the minimum-overlap edge in γ_i.
  Let $z_i = \langle x_i, \gamma_i \not\rightarrow$.
(5) For each cycle $\gamma_i \in CC$
  For each representative r_c in γ_i
   Replace r_c with $\langle\langle r_c, c\rangle\rangle$ in z_i.
(6) Concatenate the z_i to form a superstring α.

As shown in [3], α is a superstring of S. The cycle cover C formed in Step (1) identifies sets of strings that have large amounts of overlap. Steps (2)-(3) attempt to find a good way to combine the cycles from Step (1) by choosing a representative from each cycle of C and forming a second cycle cover CC on these representatives. In Step (4), each cycle of CC is broken using right extension to create a string z_i; each z_i is a superstring of the representative strings in γ_i. Step (5) performs full extension on each representative r_c; as described above, this creates a string that contains all of the strings in c as substrings. After Step (5), each of the original strings in S is a substring of one of the z_i. The concatenation of the z_i in Step (6) then correctly forms a superstring of S.

We note three details in anticipation of our algorithm presented in Section 4. In Step (2), Generic chooses representatives arbitrarily; the analysis works with any choice. In Step (4), the algorithm of [3] deletes the minimum-overlap edge in each cycle $\gamma \in CC$. In Step (5), each representative is fully extended. We show that by more carefully choosing representatives in Step (2), we can combine Steps (4) and (5) to produce a shorter superstring.

We now summarize our analysis of Generic; more details can be found in [2]. For a cycle $\gamma \in CC$, let $\mathrm{ov}_{\mathrm{cost}}(\gamma)$ be the overlap of the edge deleted in Step (4); this represents the cost of breaking the CC cycle γ to form a superstring of the representatives in γ. Let $\mathrm{Ext}(\gamma)$ be the length added in Step (6) to extend each representative in γ to include the rest of the strings in its cycle. For Generic, $\mathrm{Ext}(\gamma)$ is the cost of full extension; in general, it may be less if an algorithm requires less extension to cover the remaining strings in the cycles of C. It can be shown that

$$|\alpha| \leq \mathrm{opt}(S) + \sum_{\gamma \in CC} (\mathrm{ov}_{\mathrm{cost}}(\gamma) + \mathrm{Ext}(\gamma)) \quad . \tag{4}$$

The following key lemma from [3] is essential to this analysis:

Lemma 2.1 ([3]) *Let c and c' be cycles in a minimum cycle cover C with strings $s \in c$ and $s' \in c'$. Then the overlap between s and s' is less than $d(c) + d(c')$.*

Lemma 2.1 can be used to show that $\sum_{\gamma \in CC} \mathrm{ov}_{\mathrm{cost}}(\gamma)$ in (4) is less than $\mathrm{d}(C)$, because it provides a bound on any overlap edge in terms of $\mathrm{d}(C)$. The contribution of $\sum_{\gamma \in CC} \mathrm{Ext}(\gamma)$, the second term, is exactly $\mathrm{d}(C)$. Because the TSP is a cycle cover, $\mathrm{d}(C) \leq \mathrm{opt}(S)$, and the 3-approximation follows.

The cycle cover CC actually partitions the cycles in the cycle cover C, and hence each cycle $\gamma \in CC$ can be analyzed separately. If γ has three or more vertices, then $\mathrm{ov}_{\mathrm{cost}}(\gamma) \leq \frac{2}{3} \sum_{c \in \gamma} \mathrm{d}(c)$. Because we seek a $2\frac{2}{3}$-approximation, we can thus restrict our attention to cycles in CC that have two vertices; we refer to such cycles as 2-cycles. Given a representative $v = r_c$ for some cycle c, we use c_v to denote the cycle c of which v is a representative. We summarize this discussion with the following lemma:

Lemma 2.2 *An algorithm following the framework of the* Generic Superstring Algorithm *that, for each 2-cycle γ in CC consisting of vertices v and t, attains a bound of $ov_{\mathrm{cost}}(\gamma) + Ext(\gamma) \leq \beta(d(c_v) + d(c_t))$, for some $\beta \geq \frac{5}{3}$, is a $(1 + \beta)$-approximation algorithm for the shortest superstring problem.*

We define a few terms describing the structure of cycles. The reader is referred to [3] for a more complete discussion. We call a string s *irreducible* if all cyclic shifts of s yield unique strings, and *reducible* otherwise. We say that s has *periodicity* x if there exists a string t with $|t| = x$ such that s is substring of t^∞. Note that for a cycle c in a minimum cycle cover, $\mathrm{per}(c)$ must be irreducible; otherwise a cycle with less total prefix distance could generate the same strings, contradicting the minimality of the cycle cover. We can now state a useful corollary to Lemma 2.1:

Corollary 2.3 ([3]) *Let w be a substring of both $(\sigma_j)^\infty$ and $(\sigma_k)^\infty$ for two strings σ_j and σ_k. Then if $|w| \geq |\sigma_j| + |\sigma_k|$, at least one of σ_j or σ_k is reducible.*

3 Repeaters and their Characteristics

In the previous section, we saw that in order to obtain a better approximation for the shortest superstring problem it is sufficient to consider 2-cycles in the second cycle cover of the generic superstring algorithm. Our machinery for analyzing 2-cycles gives rise to our algorithm as well as our analysis. In this section we describe this machinery, first developed in [2].

Suppose we choose r_j and r_k as representatives of two cycles of the first cycle cover C, and they form a 2-cycle γ in CC in which one of $\mathrm{ov}(r_j, r_k)$ or $\mathrm{ov}(r_k, r_j)$ is large but the other is small. In Step (4) we will break γ to form a string, by choosing $\langle r_j, \gamma \rangle\!\!\!\rightarrow$ or $\langle r_k, \gamma \rangle\!\!\!\rightarrow$; because we are trying to maximize overlap, the obvious choice is to keep the high-overlap edge and discard the other. But if both edges have high overlap, we must still discard one of them. In a 2-cycle this will cost us up to half of the overlap, which is the worst case of the generic algorithm. We formalize the idea of a "high-overlap 2-cycle" as follows:

Definition 3.1 Let γ be a 2-cycle in the second cycle cover CC of the Generic algorithm, consisting of vertices r_j and r_k, the representatives of cycles c_j and c_k in C. Then γ is a *g-HO2-cycle* if $\min\{\mathrm{ov}(r_j, r_k), \mathrm{ov}(r_k, r_j)\} \geq g\,(\mathrm{d}(c_j) + \mathrm{d}(c_k))$ for some real $g > 0$.

In this paper we will be interested in $\frac{2}{3}$-HO2-cycles. Our strategy is to anticipate, when we choose representatives, the potential of each string to participate in a $\frac{2}{3}$-HO2-cycle. In particular

$z =$	$\underline{\text{abababrst}}$ababab	$z =$	$\underline{\text{ababadababadab}}$ababadababadabab
	) (		( ) ( ) (
$y = y_\ell =$	ababab	$y = y_\ell =$	ababadababadababa
$y = y_r =$	ababab	$y_r =$	ababadababadabab
$\sigma =$	ab	$\sigma =$	ababad
	(a)		(b)

Figure 3: Two Repeaters Per(c) is underlined. (a) shows a negative characteristic. (b) shows a positive characteristic. y and σ are also shown.

we evaluate the potential of each string to play the role of the larger-period string in the 2-cycle. Such a string must have a very specific structure; if we find a string without such a structure, we use it as the representative. Otherwise we know a great deal about the structure of the entire cycle and can trade the amount of two-way overlap against the cost of extending the representative to include the rest of the cycle.

In order to have the potential to be the larger-period string in a high-overlap 2-cycle, a string z must have a significant prefix that has some smaller period σ. In order for the high-overlap 2-cycle to occur, this smaller period σ must correspond to the period of another representative w such that $\text{ov}(w, z)$ would be large. Similarly, the suffix of z must have the same smaller period, so that $\text{ov}(z, w)$ would be large. We require some notation to describe this potential.

Definition 3.2 Let z be a string in cycle c and let σ be an irreducible string with $|\sigma| < d(c)$. Then σ is a *g-repeater of z*[1] if there exist witness strings y_ℓ and y_r, such that

1. y_ℓ is a prefix of z and y_r is a suffix of z.
2. y_ℓ and y_r are substrings of $(\sigma)^\infty$.
3. $|y_\ell|, |y_r| > g\,(\mathrm{d}(c) + |\sigma|)$.

We will always choose y_ℓ and y_r to be the maximum length prefix and suffix that satisfy conditions 1–3 above.

In this paper we will always be concerned with $\frac{2}{3}$-repeaters. Many of the definitions and results apply to g-repeaters for other values of g, but we state and prove them for $g = \frac{2}{3}$.

Consider the string z in Fig. 3b and let $g = \frac{2}{3}$. Here $\sigma = ababad$, $\text{per}(c) = ababadababadab$, $y_\ell = ababadababadababa$ and $y_r = ababadababadabab$. So $|y_\ell|, |y_r| > \frac{2}{3}(\mathrm{d}(c) + |\sigma|)$, and we say that σ is a $\frac{2}{3}$-repeater of z.

We can for simplicity define one witness y_σ which contains both y_ℓ and y_r; that is, we define y_σ to be the maximum-length substring of $(\sigma)^\infty$ that is also a substring of $\text{per}(c)^\infty$. In other words, if you take σ and try to repeat it as many times as possible, in both directions, while being consistent with c, you get y_σ. Henceforth when discussing and proving properties of cycles, we will refer to the unique maximal witness y_σ rather than to the underlying pair of witnesses y_ℓ and y_r. This simplification is conservative. When the context is clear, we will drop the σ and just refer to the witness y.

[1]This is a less general definition than that used in [1]. The original definition had a second parameter, which allows for different weights for $\mathrm{d}(c)$ and $|\sigma|$ in condition 3 of the definition. This flexibility is not required for the present result.

A witness y for some $\frac{2}{3}$-repeater σ satisfies

$$\frac{2}{3}\mathrm{d}(c) < |y| < \mathrm{d}(c) + |\sigma|. \tag{5}$$

The first inequality is immediate from Definition 3.2. The second follows from Corollary 2.3; a longer witness would have both $\mathrm{per}(c)$ and σ as periods, contradicting the Corollary 2.3.

A copy of y occurs once every $\mathrm{d}(c)$ in $\mathrm{per}(c)^\infty$. A copy of each string in c also occurs every $\mathrm{d}(c)$. Recall that we defined $\frac{2}{3}$-repeaters (Def. 3.2) in terms of some string z in a cycle c that contains witnesses y_ℓ and y_r as a prefix and suffix. In general there might be several such strings in c that could satisfy the definition. We say that a cycle c has a $\frac{2}{3}$-repeater σ if σ is a $\frac{2}{3}$-repeater of any string in c

We can now present our algorithm; additional notation and properties of $\frac{2}{3}$-repeaters are presented in Section 4.2.

4 The Algorithm

We present our algorithm G-ShortString, which is a $2\frac{2}{3}$-approximation algorithm for the shortest superstring problem. The algorithm, which we describe in Section 4.1, is built on the framework of Generic. The key to our approach is our procedure for choosing representatives, which incorporates the concepts described in Section 3.

4.1 Algorithm G-ShortString

In order to achieve a bound of $2\frac{2}{3}$ within the framework of Generic, Lemma 2.2 states that we need to concentrate on $\frac{2}{3}$-HO2-cycles. Our strategy is to anticipate, when we select a representative r_j, the possible involvement of r_j as the larger-period string in a $\frac{2}{3}$-HO2-cycle. To choose the representative for a cycle c, we evaluate a cost function for each string in c, and we select the string with the best *worst-case* cost. Our cost function resembles the desired bounds, and we explicitly attempt to minimize this function in the algorithm.

We will frequently be interested in the relationship between two substrings of $\mathrm{per}(c)^\infty$, for instance between two witness strings y and y'. As noted above, a copy of any substring of $\mathrm{per}(c)^\infty$ occurs every $\mathrm{d}(c)$ in $\mathrm{per}(c)^\infty$. We overload our notation for $\mathrm{d}_{\mathrm{pref}}(,)$ and $\mathrm{ov}(,)$ in the obvious way to refer to prefix distance $\mathrm{d}_{\mathrm{pref}}(y, y')$ and overlap $\mathrm{ov}(y, y')$. That is, if we fix any copy of y in $\mathrm{per}(c)^\infty$, $\mathrm{d}_{\mathrm{pref}}(y, y')$ is the distance from the beginning of y to the beginning of the next copy of y', and $\mathrm{ov}(y, y')$ is the overlap between the same copies. In addition, we define $\mathrm{ov}(y, y')$ for the case when y is a substring of y' or vice-versa to be the length of the shorter string. (Our original definition applied to strings from our original set of strings S, which is substring-free.) We also define the *suffix distance* $\mathrm{d}_{\mathrm{suff}}(y, y')$ to be the distance from the last character of a copy of y to the last character of the first copy of y' that ends after y.

Recall from Section 2 that Generic fully extends each representative in Step (5); that is, it concatenates each representative r_c with a copy of $\mathrm{per}(c)$. This adds length $\mathrm{d}(c)$ to the resulting superstring, and has the effect of "covering" the remaining strings in the cycle c of which r_c was the representative. Full extension has one other crucial property: it allows r_c to maintain exactly any overlap found by the algorithm in Step (3) between r_c and other strings in R. When an m-cycle in CC is broken in Step (4), such overlap must be maintained between each adjacent pair except for the overlap between the last and the first string. We observe that when $m = 2$ and we have a 2-cycle in CC consisting of vertices v and t, we need only to preserve $\mathrm{ov}(t, v)$ or $\mathrm{ov}(v, t)$, but not both. Recall that $\mathrm{Ext}(\gamma)$ is the length added by the algorithm to include the remaining strings in

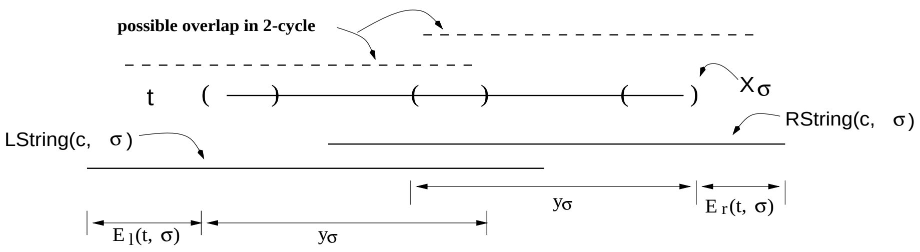


Figure 4: Definitions 4.2 and 4.3.

the cycles represented by the strings in γ. For any 2-cycle γ, our algorithm will extend only as far as is necessary to include the remaining strings in each representative's cycle, thus reducing $\mathrm{Ext}(\gamma)$ to less than the sum of the weights of the cycles in γ. The following definitions formalize this idea.

Definition 4.1 Let σ be a $\frac{2}{3}$-repeater with maximal witness y_σ in an m-vertex cycle c. Index the strings $s_1, s_2, \ldots, s_m$ so that $\mathrm{d}_{\mathrm{pref}}(y_\sigma, s_i) < \mathrm{d}_{\mathrm{pref}}(y_\sigma, s_{i+1})$, $1 \le i < m$. Then we define the *right string of* σ *in* c, $\mathrm{RString}(c, \sigma) = s_m$. The *left string of* σ *in* c, $\mathrm{LString}(c, \sigma)$ is defined symmetrically; reindex the strings s_i such that $\mathrm{d}_{\mathrm{suff}}(s_i, y_\sigma) > \mathrm{d}_{\mathrm{suff}}(s_{i+1}, y_\sigma)$, $1 \le i < m$. Then we define $\mathrm{LString}(c, \sigma) = s_1$.

In other words, if we align a copy of each of the strings in c in such a way that the first one begins as soon after the beginning of a copy of y_σ as possible, then the rightmost string is $\mathrm{RString}(c, \sigma)$. The idea is that if we choose as representative a string t, such that σ is a $\frac{2}{3}$-repeater of t, and t becomes the larger-period string in a $\frac{2}{3}$-HO2-cycle with a string v such that $\mathrm{per}(c_v) = \sigma$, then $\mathrm{RString}(c, \sigma)$ is the rightmost string that we will have to include if we extend v to the right. Figure 4 illustrates Definitions 4.1 and 4.2.

Definition 4.2 Let σ be a $\frac{2}{3}$-repeater of a string t in cycle c. Then the *right* σ*-extension with respect to* t, $\mathrm{E}_r(t, \sigma) = \mathrm{d}_{\mathrm{suff}}(y_\sigma, \mathrm{RString}(c, \sigma))$. The *left* σ*-extension with respect to* t, $\mathrm{E}_\ell(t, \sigma) = \mathrm{d}_{\mathrm{pref}}(\mathrm{LString}(c, \sigma), y_\sigma)$.

For each string t in cycle c_t, we are interested in the worst-case cost of choosing t as the representative of c_t. In particular, if σ is a $\frac{2}{3}$-repeater of t, we wish to calculate the cost of choosing t and having t involved in a $\frac{2}{3}$-HO2-cycle γ with some string v such that $\mathrm{per}(c_v) = \sigma$. By Lemma 2.2 we need to bound $\mathrm{ov}_{\mathrm{cost}}(\gamma) + \mathrm{Ext}(c_t) + \mathrm{Ext}(c_v)$. Suppose without loss of generality that we choose to keep the 2-cycle edge (v, t); then we will extend v to the left and t to the right. Let y_r be the suffix of t that is the witness string for σ. From the end of y_r, we need to extend to the right only as far as necessary to include $\mathrm{RString}(c, \sigma)$. We also have to extend v to include the remaining strings in c_v; we assume the cost of full extension, which is $\mathrm{d}(c_v)$. This motivates the following definition.

Definition 4.3 Let σ be a $\frac{2}{3}$-repeater of string t in cycle c. Then the *anticipated cost* of choosing t as representative and forming a 2-cycle with a string with period σ is

$$\mathrm{Cost}(t, \sigma) = |y_\sigma| + \min\{\mathrm{E}_\ell(t, \sigma), \mathrm{E}_r(t, \sigma)\} + |\sigma| \ .$$

What we seek, then, is to *minimize*, in our choice of representative t, the *maximum* over all $\frac{2}{3}$-repeaters of t, the anticipated cost $\mathrm{Cost}(t, \sigma)$. We imagine that once we choose the representative

r of cycle c, an adversary will match r with another representative v whose period $\mathrm{per}(c_v)$ matches some $\frac{2}{3}$-repeater in r to form a high-overlap 2-cycle. Allowing $\sigma \in t$ to mean "σ a $\frac{2}{3}$-repeater of t", we seek [2]

$$\mathrm{BestRep}(c) = \mathrm{argmin}_{t \in c} \left\{ \max_{\sigma \in t} \{\mathrm{Cost}(t, \sigma)\} \right\} .$$

Procedure G-FindReps(c), shown below, calculates the anticipated cost for each pair (t, σ) such that t is a string in c and σ is a $\frac{2}{3}$-repeater of t.

Procedure G-FindReps(c_j)

(1) Find all $\frac{2}{3}$-repeaters and associated characteristics in c_j.
(2) **If** any string t has no $\frac{2}{3}$-repeaters
    Then $r_j = t$;
    Else $r_j = \mathrm{BestRep}(c_j)$;
(3) **Return** r_j.

The main body of G-ShortString resembles Generic, except that representatives are selected in Step (2) by a call to procedure G-FindReps(c), and our Step (4) combines Steps (4) and (6) of Generic.

Algorithm G-ShortString

(1) Form minimum cycle cover C on distance graph G.
(2) For each cycle $c \in C$
    Call G-FindReps(c) to choose representative r_c; Add r_c to R.
  Let G' be the subgraph induced by R.
(3) Form minimum cycle cover CC on G'.
(4) For each cycle γ_i in CC:
    if γ_i is a $\frac{2}{3}$-HO2-cycle (v, t)
  (a) **then if** $\mathrm{ov}(t, v) + \mathrm{E}_r(t, \mathrm{per}(c_v)) \leq \mathrm{ov}(v, t) + \mathrm{E}_\ell(t, \mathrm{per}(c_v))$
      then let $z_i = \langle \twoheadleftarrow\!\langle v, c_v\rangle, \langle t, c_t \rangle\!\!\twoheadrightarrow \rangle$;
      else let $z_i = \langle \twoheadleftarrow\!\langle t, c_t\rangle, \langle v, c_v \rangle\!\!\twoheadrightarrow \rangle$;
  (b) **else** let (w_i, x_i) be the least-overlap edge in γ_i;
      Let $z_i = \langle x_i, \gamma_i \rangle\!\!\twoheadrightarrow$;
      For each string u in γ_i
        Replace u with $\langle\langle u, c_u \rangle\rangle$.
(5) Concatenate the strings z_i from (4) to form superstring α.

Step (4b) above, which applies to non-$\frac{2}{3}$-HO2-cycle, is simply the Generic algorithm; each CC cycle is broken by deleting the least-overlap edge, and each representative is fully extended to include the other strings in its cycle. In Step (4a) we handle $\frac{2}{3}$-HO2-cycles. We can either have v to the left of t, or vice-versa. In either case, we use left extension on the left string and right extension on the right string, thus extending only as far as necessary to include the remaining strings in c_v and c_t.

The algorithm G-ShortString correctly computes a superstring of the set of strings S. This follows from the correctness of Generic. Our method of choosing representatives for each cycle is a special case of the method of Generic, which chooses an arbitrary string as representative. In step (4b), we do exactly what Generic does. In step (4a), we use a different criterion for breaking a cycle $\gamma \in CC$, and we only extend each representative far enough to "cover" all of the strings in its cycle. Each string is therefore included in the solution α.

G-ShortString runs in polynomial time. The distance graph G can be built in $O(|S| + n^2)$ time [11], and the cycle cover computations take $O(n^3)$ time [17]. These two results determine the

[2] The *argmin* is the argument that achieves the minimum value. For instance, let $x = 5$ and $y = 3$. Then $\min(x, y)$ is 3, while $\mathrm{argmin}(x, y)$ is y.

running time of GENERIC, $O(|S| + n^3)$. In addition, our procedure G-FINDREPS(c) must find all of the $\frac{2}{3}$-repeaters in each cycle $c \in C$. This can be done naively in polynomial time by examining a prefix and suffix of each string and determining, for each j, $2 \leq j < \mathrm{d}(c)$, whether the prefix and suffix have periodicity j.

In order to analyze the length bound attained by G-SHORTSTRING, we require some additional notation and properties of $\frac{2}{3}$-repeaters.

4.2 Properties of Strings with $\frac{2}{3}$-Repeaters

We begin by defining some terminology and establishing some simple properties of $\frac{2}{3}$-repeaters.

A *small* $\frac{2}{3}$-repeater in a cycle c is a $\frac{2}{3}$-repeater such that $\frac{2}{3}\left(\mathrm{d}(c) + |\sigma|\right) < \mathrm{d}(c)$, or equivalently if $|\sigma| < \frac{1}{2}\mathrm{d}(c)$. When discussing more than one $\frac{2}{3}$-repeater, say σ and σ', we shall often use the relative terms such as *smaller*, *smallest*, *larger*, *largest*; we shall always be referring to the relationship between $|\sigma|$ and $|\sigma'|$. There may be several small $\frac{2}{3}$-repeaters in a cycle, and we are able to bound the number of small $\frac{2}{3}$-repeaters of a string.

Lemma 4.4 *Let s be a string in a cycle c. Then there can be at most one small $\frac{2}{3}$-repeater of s.*
Proof: Suppose for purpose of contradiction that there exist two such small $\frac{2}{3}$-repeaters σ and σ'. Let $y_\ell(\sigma)$ and $y_\ell(\sigma')$ be the prefixes of s that are the left witness strings of σ and σ' respectively. Let $y_\ell = \mathrm{argmin}\{|y_\ell(\sigma)|, |y_\ell(\sigma')|\}$ be the prefix of s which is periodic in both σ and σ'. Applying Corollary 2.3, Definition 3.2, and the fact that $|\sigma'| < \frac{1}{2}\mathrm{d}(c)$, we get

$$|\sigma| > |y_\ell| - |\sigma'| > \frac{2}{3}\mathrm{d}(c) - \frac{1}{3}|\sigma'| > \frac{1}{2}\mathrm{d}(c) \ ,$$

a contradiction because σ is a small $\frac{2}{3}$-repeater. ■

We will frequently be interested in relating the beginning and ends of the strings in c to the witnesses y in c. The following definition gives us a point of reference for each witness.

Definition 4.5 Let c be a cycle with g-repeater σ and maximal witness y. Fix a copy of y in $\mathrm{per}(c)^\infty$. The point just to the left of the first character of y is the *head of y*. Index this point as 0 and continue the indices between each character leftward and rightward to cover the interval $[-\mathrm{d}(c)..\mathrm{d}(c)]$. Now mark the point $|y| - \mathrm{d}(c)$ and call it the *tail* of σ. The *characteristic* of σ, X_σ, is the interval from the head to the tail. If $|y| - \mathrm{d}(c) > 0$ we call $[0..|y| - \mathrm{d}(c)]$ a *positive characteristic* X_σ. If $|y| - \mathrm{d}(c) \leq 0$ we call $[|y| - \mathrm{d}(c)..0]$ a *negative characteristic* X_σ.

We can picture the characteristics of the repeaters of a cycle c in terms of parentheses. Fig. 3b illustrates this idea for positive characteristics. The left and right ends of y_σ are marked with left and right parentheses; these correspond to the head and tail of adjacent copies of X_σ.

A negative characteristic appears in Fig. 3a and can be pictured as a single solid entity (perhaps of size zero) that spans the gap between copies of y. In this example rst is the negative characteristic. Each characteristic appears once every $\mathrm{d}(c)$. Intuitively, the characteristic of a repeater borders the portion of $\mathrm{per}(c)$ that must be included as a prefix and suffix of some string z if z is to participate in a high-overlap 2-cycle.

We say that two characteristics X_{σ_i}, X_{σ_j} are *nested* if X_{σ_i} is a positive characteristic and X_{σ_j} falls within X_{σ_i}. We say that two characteristics X_{σ_i}, X_{σ_j} are *disjoint* if their intervals are disjoint. Otherwise we say that X_σ and $X_{\sigma'}$ are *linked.*

The following lemma gives us a lower bound on the size of a $\frac{2}{3}$-repeater whose characteristic has the characteristic of another $\frac{2}{3}$-repeater nested within it; such a $\frac{2}{3}$-repeater cannot be small.

Lemma 4.6 *Let X_σ be a positive characteristic in cycle c and $X_{\sigma'}$ a characteristic nested within X_σ with $|\sigma| > |\sigma'|$. Then $|\sigma| > \frac{1}{2}d(c)$.*

Proof: In this case the witness y' is completely contained within the witness y. We apply Corollary 2.3 and the definition of $\frac{2}{3}$-repeater to get

$$\begin{aligned} |\sigma| + |\sigma'| &> \mathrm{ov}(y, y') \\ &= |y'| \\ &> \frac{2}{3}(\mathrm{d}(c) + |\sigma'|) \\ \Rightarrow |\sigma| &> \frac{2}{3}\mathrm{d}(c) - \frac{1}{3}|\sigma'|, \end{aligned}$$

which implies $|\sigma| > \frac{1}{2}\mathrm{d}(c)$ because $|\sigma| > |\sigma'|$. ■

We are able to prove Lemmas 4.4 and 4.6 using only the size of the $\frac{2}{3}$-repeaters. The remaining technical results in this section require more machinery relating the locations of the $\frac{2}{3}$-repeaters.

In [2] we showed that $\frac{3}{4}$-repeaters are *well parenthesized*; that is, that the characteristics of $\frac{3}{4}$-repeaters may be disjoint or nested, but never linked. The proof of this strong structural characterization was nontrivial, and very sensitive to the value of $g = \frac{3}{4}$. We have been able neither to prove this property for $g = \frac{2}{3}$, nor to generate an example of non-well-parenthesized $\frac{2}{3}$-repeaters. Because $\frac{2}{3}$-repeaters may not be well parenthesized, we will often be faced in our analysis with situations in which two positive characteristics are linked, as pictured in Figure 5. (Recall that two positive characteristics are linked if they overlap, but neither contains the other.) The following lemma and its corollary gives us strong bounds on the size of the two $\frac{2}{3}$-repeaters and on their difference. In order to prove the lemma, we require a proof technique introduced in [2], the *shift argument*. We describe this technique below.

We apply the shift argument to cycles that include two or more repeaters. We are generally interested in proving that some property holds; we assume that it does not, and use the shift argument to derive a contradiction. We begin with the following observation, which can easily be verified by the definition of maximal witness.

Observation 4.7 ([2]) *Let y be the maximal witness for a g-repeater σ in a cycle c, and fix a copy y^* of y in $per(c)^\infty$. Index the character positions of $per(c)^\infty$ with the character to the left of y^* as 0, the first character of y^* as 1, and continuing to the right beyond the end of y^*. Let Char(i) be the character in position i. Then*

$$a)\, Char(0) \neq Char(|\sigma|)$$

and

$$b)\, Char(|y^*| - |\sigma| + 1) \neq Char(|y^*| + 1).$$

In each shift argument our goal will be to show that either inequality a) or b) in Observation 4.7 is violated and the terms are indeed equal. We will do so by making a series of *shifts* between characters, which we know to be identical, by the periodic structure of the strings. In particular, within any copy of y, any two characters that are σ apart are identical, and in $\mathrm{per}(c)^\infty$, any two characters that are $\mathrm{d}(c)$ apart are identical. We call such shifts *valid*. We will begin at either the character immediately preceding or following a copy of y or y', and perform a series of shifts which will bring us to the position whose character is supposed to be unequal. If these shifts are valid, then the two characters must be equal, contradicting our initial assumption that the characteristics X_σ and $X_{\sigma'}$ could overlap.

We introduce notation to describe the sequence of shifts. We give a starting position and a position at which we wish to arrive, relative to the starting position. We also give the series of

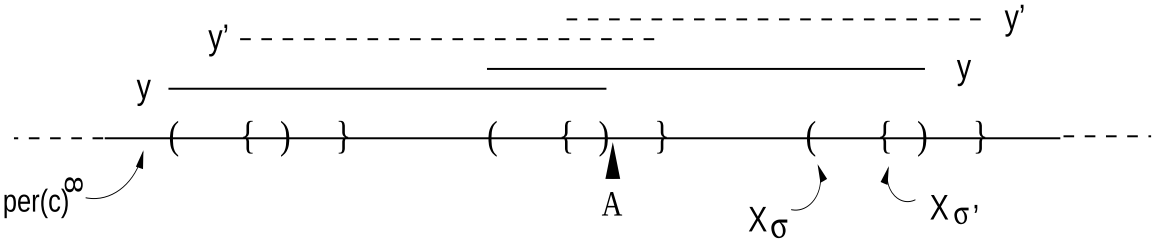


Figure 5: The characteristics X_σ and $X_{\sigma'}$ are linked, as in Lemma 4.8.

moves and a set of requirements, that is, conditions on the various parameters that must be met in order for the moves to all be valid. Below the box, we show that the conditions for validity are indeed satisfied, which gives us a contradiction for the region in which the shifts are valid.

Lemma 4.8 *Let σ and σ' be two $\frac{2}{3}$-repeaters with positive characteristics in a cycle c, with $|\sigma| > |\sigma'|$, and X_σ and $X_{\sigma'}$ linked. Let $k = \left\lfloor \frac{|\sigma|}{|\sigma'|} \right\rfloor$. Then $|\sigma| - k|\sigma'| > |y_\sigma| - d(c)$.*

Proof: We apply the following shift argument, using start position (A) in Figure 5:

Start: (A)			**Goal:** $-\lvert\sigma\rvert$
No.	*Move*	*Requirement*	*Comments*
1.	$+k\lvert\sigma'\rvert$	$k\lvert\sigma'\rvert < \mathrm{d}(c)$	$k\lvert\sigma'\rvert < \lvert\sigma\rvert < \mathrm{d}(c)$
2.	$-\lvert\sigma\rvert$	$\lvert\sigma\rvert - k\lvert\sigma'\rvert \le \lvert y_\sigma\rvert - \mathrm{d}(c)$	See below.
3	$-k\lvert\sigma'\rvert$	$\lvert\sigma\rvert < \mathrm{d}(c)$	Def. Repeater.

Because move #2 is the only one whose validity is conditional, we conclude the negation of that condition, i.e. $|\sigma| - k|\sigma'| > |y_\sigma| - \mathrm{d}(c)$. ■

Corollary 4.9 *Let σ and σ' be two $\frac{2}{3}$-repeaters with positive characteristics in a cycle c, with $|\sigma| > |\sigma'|$, and X_σ and $X_{\sigma'}$ linked. Let $k = \left\lfloor \frac{|\sigma|}{|\sigma'|} \right\rfloor$. Then $|\sigma'| > |y_\sigma| - d(c)$.*

Proof: By the choice of k and Lemma 4.8,

$$|\sigma'| > |\sigma| - k|\sigma'| > |y_\sigma| - \mathrm{d}(c).$$

■

In our analysis, we will be interested in *lower bounds* on the size of potentially small $\frac{2}{3}$-repeaters in terms of some measure of distance that will correspond to extension cost. The following two lemmas provides such bounds for two important cases in which three characteristics are involved. Our choice of dimensions for identifying the relative positions of the three characteristics may seem unnatural now, but will simplify our task in Section 4.3.

Lemma 4.10 *Let σ, σ' and σ'' be $\frac{2}{3}$-repeaters in cycle c, with $X_{\sigma'}$ and $X_{\sigma''}$ disjoint, and with $X_{\sigma'}$ nested to the left of $X_{\sigma''}$ within X_σ, Then $|\sigma'| > d_{\mathrm{pref}}(y', y) + |y| - 2d(c)$ and $|\sigma''| > d_{\mathrm{suff}}(y, y'') + |y| - 2d(c)$.*

Proof: Figure 6 illustrates the start positions of our shift arguments.

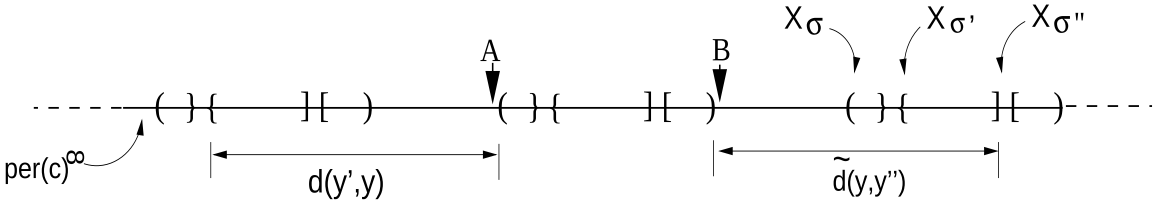


Figure 6: Proof of Lemma 4.10.

Start: (A)			**Goal:** $+\lvert\sigma\rvert$
No.	*Move*	*Requirement*	*Comments*
1.	$+\lvert\sigma''\rvert$	$\lvert\sigma''\rvert \le \mathrm{d}_{\mathrm{suff}}(y, y'') + \lvert y\rvert - 2\mathrm{d}(c)$	See below.
2.	$+\lvert\sigma\rvert$	$\lvert\sigma\rvert + \lvert\sigma''\rvert < \lvert y\rvert - \mathrm{d}(c) + \mathrm{d}_{\mathrm{suff}}(y, y'')$	See below.
3	$-\lvert\sigma''\rvert$	$\lvert\sigma\rvert + \lvert\sigma''\rvert > \lvert y\rvert - \mathrm{d}(c) + \lvert\sigma''\rvert$	See below.

Start: (B)			**Goal:** $-\lvert\sigma\rvert$
No.	*Move*	*Requirement*	*Comments*
1.	$-\lvert\sigma'\rvert$	$\lvert\sigma'\rvert \le \mathrm{d}_{\mathrm{pref}}(y', y) + \lvert y\rvert - 2\mathrm{d}(c)$	See below.
2.	$-\lvert\sigma\rvert$	$\lvert\sigma\rvert + \lvert\sigma'\rvert < \lvert y\rvert - \mathrm{d}(c) + \mathrm{d}_{\mathrm{pref}}(y', y)$	See below.
3	$+\lvert\sigma'\rvert$	$\lvert\sigma\rvert + \lvert\sigma'\rvert > \lvert y\rvert - \mathrm{d}(c) + \lvert\sigma'\rvert$	See below.

Requirement #3 for both of the above is always true because $|\sigma| > |y| - \mathrm{d}(c)$ by Eq. 5. Because $|\sigma| < \mathrm{d}(c)$, #1 implies #2; therefore #1 must be false, and we have

$$
\begin{aligned}
|\sigma''| &> \mathrm{d}_{\mathrm{suff}}(y, y'') + |y| - 2\mathrm{d}(c) \\
|\sigma'| &> \mathrm{d}_{\mathrm{pref}}(y', y) + |y| - 2\mathrm{d}(c).
\end{aligned}
$$

■

Lemma 4.11 *Let σ, σ' and σ'' be $\frac{2}{3}$-repeaters in cycle c, with maximal witnesses y, y' and y''. Let $|\sigma| > |\sigma'| > |\sigma''|$, and X_σ and $X_{\sigma'}$ positive. If $X_{\sigma'}$ is linked with X_σ, then $\min\{d_{\mathrm{pref}}(y', y), d_{\mathrm{suff}}(y, y'')\} < \frac{5}{3}d(c) + \frac{2}{3}|\sigma| - |y|$.*

Proof: By the condition of the lemma we know that X_σ and $X_{\sigma'}$ are linked, but we do not know the relationship between between $X_{\sigma''}$ and the other two characteristics. The characteristic $X_{\sigma''}$ may be nested within one or both of X_σ and $X_{\sigma'}$ as in Figure 7(a) or (b), or it may be linked with one or both of X_σ and $X_{\sigma'}$ as in Figure 7 (c) or (d). In any of these cases we can apply Corollary 2.3 to the overlap between y' and y'':

$$|\sigma'| + |\sigma''| > \mathrm{d}_{\mathrm{pref}}(y', y) + \mathrm{d}_{\mathrm{suff}}(y, y'') + |y| - 2\mathrm{d}(c),$$

which implies

$$|\sigma'| > \frac{1}{2}(\mathrm{d}_{\mathrm{pref}}(y', y) + \mathrm{d}_{\mathrm{suff}}(y, y'')) + \frac{1}{2}|y| - \mathrm{d}(c). \tag{6}$$

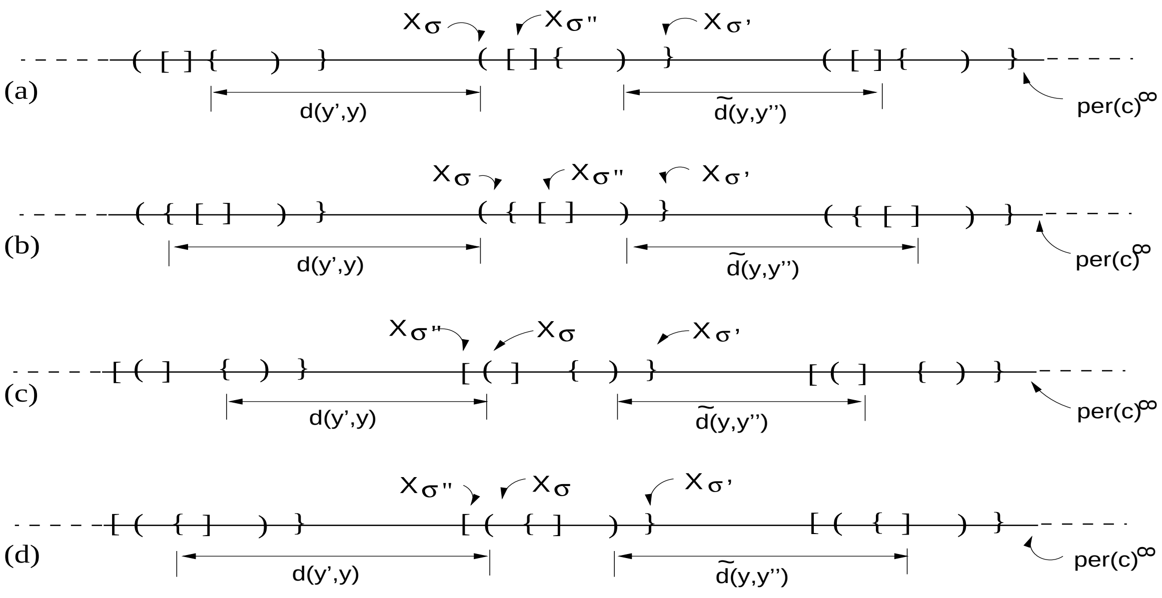


Figure 7: Lemma 4.11: $X_{\sigma'}$ is linked with X_σ. $X_{\sigma''}$ may be nested within $X_{\sigma'}$ (b), or only nested within X_σ(a). If $X_{\sigma''}$ is linked with X_σ, it may or may not also be linked with $X_{\sigma'}$ ((d) and (c) respectively).

We now use Lemma 4.8 and Equation 6 to obtain

$$\begin{aligned}
|\sigma| &> |\sigma'| + |y| - \mathrm{d}(c) \\
&> \frac{1}{2}(\mathrm{d}_{\mathrm{pref}}(y', y) + \mathrm{d}_{\mathrm{suff}}(y, y'')) + \frac{1}{2}|y| - \mathrm{d}(c) + |y| - \mathrm{d}(c) \\
&= \frac{1}{2}(\mathrm{d}_{\mathrm{pref}}(y', y) + \mathrm{d}_{\mathrm{suff}}(y, y'')) + \frac{3}{2}|y| - 2\mathrm{d}(c).
\end{aligned}$$

Solving for $\frac{1}{2}(\mathrm{d}_{\mathrm{pref}}(y', y) + \mathrm{d}_{\mathrm{suff}}(y, y''))$ and using Definition 3.2 gives us our result:

$$\begin{aligned}
\frac{1}{2}(\mathrm{d}_{\mathrm{pref}}(y', y) + \mathrm{d}_{\mathrm{suff}}(y, y'')) &< |\sigma| - \frac{3}{2}|y| + 2\mathrm{d}(c) \\
&< |\sigma| - |y| + 2\mathrm{d}(c) - \frac{1}{2}(\frac{2}{3}\mathrm{d}(c) + \frac{2}{3}|\sigma|) \\
&= \frac{5}{3}\mathrm{d}(c) + \frac{2}{3}|\sigma| - |y|.
\end{aligned}$$

■

4.3 Proof of the Length Bound

We now analyze our algorithm G-ShortString. As noted in Section 2, we can consider each cycle in CC independently. Lemma 4.12 addresses the case of non-$\frac{2}{3}$-HO2-cycles. We partition the instances of $\frac{2}{3}$-HO2-cycles and consider them in Lemmas 4.13 and 4.15.

Lemma 4.12 *For each cycle $\gamma \in CC$ that is not a $\frac{2}{3}$-HO2-cycle, Algorithm* G-ShortString *produces a superstring no longer than* Generic *would produce on the same cycle γ.*

Proof: Step (4b) of G-ShortString handles any cycle $\gamma \in CC$ that is not a $\frac{2}{3}$-HO2-cycle. It selects the least-overlap edge in γ for deletion in order to construct a superstring of the representatives in γ, as does Generic. It then fully extends each representative $r \in \gamma$ by the length $\mathrm{d}(c_r)$, as does Generic, to include the remaining strings in c_r. ∎

We now must show that for each $\frac{2}{3}$-HO2-cycle, we attain the bound specified by Lemma 2.2. That is, for each $\frac{2}{3}$-HO2-cycle γ consisting of representatives t and v, we need to show that $\mathrm{ov}_{\mathrm{cost}}(\gamma) + \mathrm{Ext}(\gamma) \leq \frac{5}{3}(\mathrm{d}(c_t) + \mathrm{d}(c_v))$. By Definition 4.3, this is equivalent to showing that $\mathrm{Cost}(t, \mathrm{per}(c_v)) \leq \frac{5}{3}(\mathrm{d}(c_t) + \mathrm{d}(c_v))$. When Cost(,) was applied in BestRep(c_t), the $\frac{2}{3}$-repeaters in each string in c_t were identified. String t must have been chosen as representative, and we are now considering the case in which t has a $\frac{2}{3}$-repeater that matches $\mathrm{per}(c_v)$ for some cycle c_v. Lemmas 4.13 and 4.15, respectively, handle the case when all strings in c_t have a small $\frac{2}{3}$-repeater, and when at least one string in c_t has no small $\frac{2}{3}$-repeaters.

Lemma 4.13 *Let γ be a $\frac{2}{3}$-HO2-cycle in CC with r_j the representative of cycle c_j and r_k the representative of c_k, let $d(c_j) \geq d(c_k)$, and let all strings in c_j have at least one small $\frac{2}{3}$-repeater. Then $ov_{\mathrm{cost}}(\gamma) + Ext(\gamma) \leq \frac{5}{3}(d(c_j) + d(c_k))$.*

Proof: We will consider all of the strings in c_j and the choice of r_j as representative. The proof of Lemma 4.4 suggests our strategy: if two strings with different $\frac{2}{3}$-repeaters begin near each other, then the sum of their periods must be close to $\mathrm{d}(c_j)$. If they do not begin near each other, then we can save on extension by the amount of this gap.

By assumption, each string has at least one small $\frac{2}{3}$-repeater. No string has more than one small $\frac{2}{3}$-repeater by Lemma 4.4, and so each string has exactly one small $\frac{2}{3}$-repeater. More than one string may have the same small $\frac{2}{3}$-repeater.

Proposition 4.14 *Let σ and σ' be small $\frac{2}{3}$-repeaters in cycle c. Let Q be the set of strings that have σ as a $\frac{2}{3}$-repeater, and let Q' be the set of strings that have σ' as a $\frac{2}{3}$-repeater. Then there is a rotation of the cyclic ordering of the strings in c such that all of the strings in Q appear before all of the strings in Q'.*

Proof: For purpose of contradiction let t and v be two strings in Q and let t' and v' be two strings in Q' such that they appear in the cyclic order t, t', v, v'. Without loss of generality let $\mathrm{d}_{\mathrm{pref}}(t, v) \leq \frac{1}{2}\mathrm{d}(c)$; otherwise $\mathrm{d}_{\mathrm{pref}}(v, t) \leq \frac{1}{2}\mathrm{d}(c)$ and the same argument follows. Consider the prefixes of t and v which are the left witness for σ; both prefixes must be substrings of the same copy of y_σ. Since t' is between t and v, then it also must have a prefix y'_ℓ which has period σ. The same argument holds for the suffixes of t, v and t', so σ must be a $\frac{2}{3}$-repeater of t'. But then t' has both σ and σ', contradicting Lemma 4.4. □

We resume our analysis. Let σ_1 be the largest of the small $\frac{2}{3}$-repeaters in c_j, and let Q_1 be the set of strings that have σ_1. Number the remaining small $\frac{2}{3}$-repeaters $\sigma_2, \ldots, \sigma_m$, and let Q_i, $1 \leq i \leq m$, be the set of strings of which σ_i is a $\frac{2}{3}$-repeater. The Q_i's partition the strings of the cycle, and by Claim 4.14 the Q_i's form a cyclic ordering. Let u_i, $1 \leq i \leq m$ be the leftmost string in each group Q_i, and let w_i, $1 \leq i \leq m$ be the rightmost string in each group Q_i. Let $\ell_i = \mathrm{d}_{\mathrm{pref}}(u_i, u_{i+1})$, $1 \leq i < j$. (See Fig. 8.)

First we apply Corollary 2.3 to derive a lower bound on ℓ_1:

$$|\sigma_1| + |\sigma_2| > \mathrm{ov}(y_{\sigma_1}, y_{\sigma_2}) \geq |y_{\sigma_1}| - \ell_1 \ ,$$

which implies that $2|\sigma_1| > |y_{\sigma_1}| - \ell_1$, or

$$\ell_1 > |y_{\sigma_1}| - 2|\sigma_1| \ . \tag{7}$$

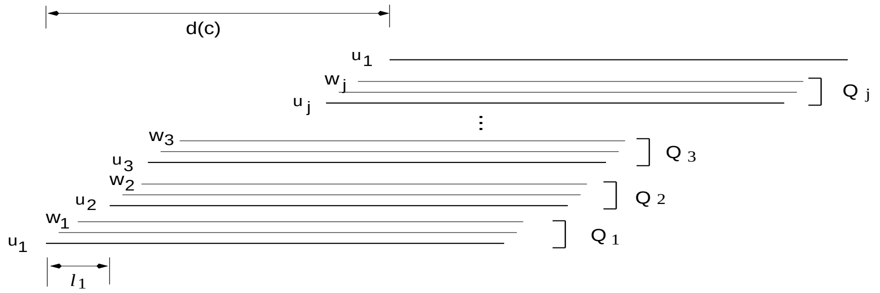


Figure 8: Cycle c_j in Lemma 4.13.

Now we bound the cost incurred by breaking γ and extending the resulting string to include the other strings in c_j and c_k:

$$\begin{aligned} \mathrm{ov}_{\mathrm{cost}}(\gamma) + \mathrm{Ext}(\gamma) &= \mathrm{Cost}(u_1, \sigma_1) \\ &= |y_{\sigma_1}| + \min\{\mathrm{E}_\ell(u_1,\sigma_1), \mathrm{E}_r(u_1,\sigma_1)\} + |\sigma_1| \\ &\le |y_{\sigma_1}| + \mathrm{E}_\ell(u_1,\sigma_1) + |\sigma_1| \ . \end{aligned}$$

If we extend u_1 to the left, the last string we will have to cover will be u_2, so $\mathrm{E}_\ell(u_1,\sigma_1) = \mathrm{d}(c) - \ell_1$, and then we use (7):

$$\begin{aligned} \mathrm{Cost}(u_1,\sigma_1) &\le |y_{\sigma_1}| + \mathrm{d}(c) - \ell_1 + |\sigma_1| \\ &\le \mathrm{d}(c) + 3|\sigma_1| \\ &\le \frac{5}{3}(\mathrm{d}(c) + |\sigma_1|) \ . \end{aligned}$$

The last inequality follows from the fact that σ_1 is a small $\frac{2}{3}$-repeater, so $|\sigma_1| < \frac{1}{2}\mathrm{d}(c)$. ■

We have now shown that the bound holds when there is a small $\frac{2}{3}$-repeater in each string in c_t. The following lemma handles the case when at least one string in c_t has no small $\frac{2}{3}$-repeaters.

Lemma 4.15 *Let γ be a $\frac{2}{3}$-HO2-cycle in CC with r_j the representative of cycle c_j and r_k the representative of c_k, let $d(c_j) \ge d(c_k)$, and let there be at least one string in c_j that does not have a small $\frac{2}{3}$-repeater. Then $ov_{\mathrm{cost}}(\gamma) + Ext(\gamma) \le \frac{5}{3}(d(c_j) + d(c_k))$.*

Proof: Throughout the proof of this lemma, we fix s to be a particular string in c_j; in some cases, but not all, s will prove to be a good choice of representative. When it does not, we will show that there is another string whose anticipated cost is small enough, and therefore would have been chosen in FINDREPS(c_j).

Let A be the set of m' strings that do not have a small repeater; there is at least one such string by assumption. In order to identify s, rename the strings in A, $a_1, \ldots, a_{m'}$. Let σ_i be the smallest repeater of string a_i. Then let $s = a_k$, with k chosen such that $|\sigma_k| \ge |\sigma_i|$, $1 \le i \le m'$. In other words, s is the string whose smallest $\frac{2}{3}$-repeater is the largest, over all the strings in c_j.

Our analysis has two main cases, which depend on the composition of the cycle c_j. In the first of these, we can show that s is a good choice of representative; that is, the worst-case cost of choosing s is within our bounds. In the remaining cases, we show that s is sometimes a good choice; if it is *not* a good choice, there must be some particular string s' that is a good choice.

Case 1: $\min\{\mathrm{E}_\ell(s,\sigma), \mathrm{E}_r(s,\sigma)\} \le \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$.

If $\mathrm{E}_\ell(s,\sigma) \le \mathrm{E}_r(s,\sigma)$, then we extend s to the left; otherwise, we extend s to the right to cover the

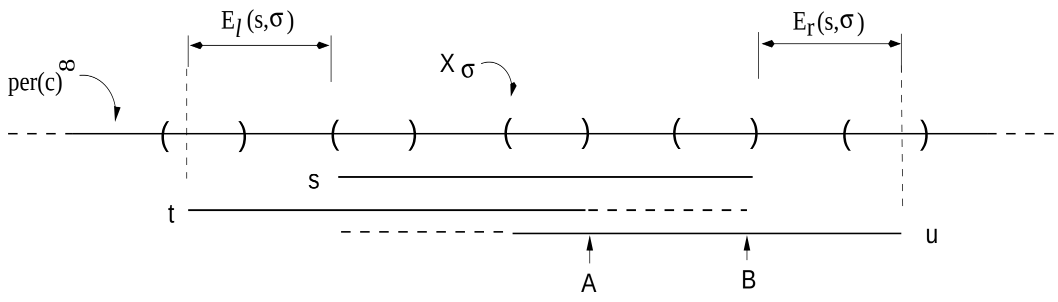


Figure 9: Case 2 of Lemma 4.13. Determining the range of possible t and u.

remaining strings in c. We bound the cost incurred by γ:

$$\begin{aligned}
\mathrm{ov}_{\mathrm{cost}}(\gamma) + \mathrm{Ext}(\gamma) &= \mathrm{Cost}(s,\sigma) \\
&\leq |y_\sigma| + \min\{\mathrm{E}_\ell(s,\sigma), \mathrm{E}_r(s,\sigma)\} + |\sigma| \\
&\leq |y_\sigma| + \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma| + |\sigma| \\
&= \frac{5}{3}(\mathrm{d}(c_j) + |\sigma|).
\end{aligned}$$

This concludes the analysis of Case 1.

Case 2: $\min\{\mathrm{E}_\ell(s,\sigma), \mathrm{E}_r(s,\sigma)\} > \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$.
If Case 1 does not apply, then, as in Figure 9, there must be a string $t = \mathrm{LString}(c,\sigma)$ and a string $u = \mathrm{RString}(c,\sigma)$, not necessarily distinct, that extend to the left and right, respectively, too far for s to be extended within the bounds of Case 1. In particular, let X^ℓ_σ and X^r_σ be the copies of X_σ in which s begins and ends. Then t must extend into X^ℓ_σ, because otherwise $\mathrm{E}_\ell(s,\sigma) \leq 2\mathrm{d}(c_j) - |y_\sigma| \leq \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$, since $|\sigma| > \frac{1}{2}\mathrm{d}(c_j)$. We also note that t cannot extend to the left beyond X^ℓ_σ, or we could simply shift it over $\mathrm{d}(c_j)$ to the right. Therefore the left end of t is in X^ℓ_σ. The right end of t must also be within $\mathrm{d}(c_j)$ of the right end of s, or between points A and B marked in Figure 9. Similarly, the right end of u is in X^r_σ, and the left end may be anywhere within $\mathrm{d}(c_j)$ to the right of the left end of s.

Because each string in c must have at least one $\frac{2}{3}$-repeater, let σ' be the smallest $\frac{2}{3}$-repeater of t, and σ'' the smallest $\frac{2}{3}$-repeater of u. The position of the right end of t (left end of u) will determine whether $X_{\sigma'}$ ($X_{\sigma''}$) is nested within X_σ or linked with it. The remaining cases which we consider all have $\min\{\mathrm{E}_\ell(s,\sigma), \mathrm{E}_r(s,\sigma)\} > \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$ and are determined by whether $t = u$ and whether $X_{\sigma'}$ and $X_{\sigma''}$ are linked with or nested within X_σ.

In order to simplify our analysis, we will often assume that a string with a repeater σ extends from the left end of one copy of y_σ to the right end of another copy of y_σ. This assumption is pessimistic in two ways; first, we may be over-charging for extension, if a string does not go as far as the right end of y_σ and we assume it does. Second, witnesses longer than the minimum for $\frac{2}{3}$-repeaters give us stronger results when we apply Corollary 2.3.

Case 2a: $\min\{\mathrm{E}_\ell(s,\sigma), \mathrm{E}_r(s,\sigma)\} > \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$, $t = u$.
We will show that t can be extended within the desired bounds. Recall that σ' is the smallest $\frac{2}{3}$-repeater of t. Observe that $\mathrm{E}_\ell(s,\sigma)$ and $\mathrm{E}_r(s,\sigma)$ span the length of a single copy of $y_{\sigma'}$ with some overlap between two copies of X_σ. This observation gives rise to the following identity:

$$\mathrm{E}_\ell(s,\sigma) + \mathrm{E}_r(s,\sigma) = |y_{\sigma'}| \ + 2\mathrm{d}(c_j) - |y_\sigma|. \tag{8}$$

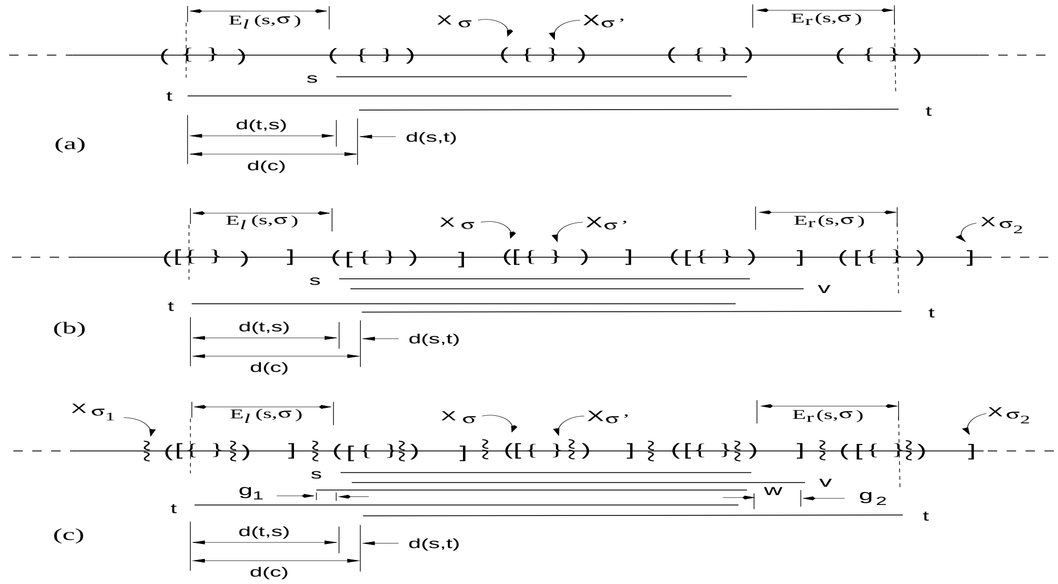


Figure 10: Case 2A of Lemma 4.15. (a) Without an interloper. (b) With one interloper v. (c) With two interlopers v and w.

Now consider extending t to the right. Any string t' which begins within $\mathrm{d}_{\mathrm{pref}}(t,s)$ of the beginning of t must end before s due to the no-substring assumption, and we will only need to extend t by $\mathrm{d}_{\mathrm{suff}}(y_{\sigma'}, y_\sigma)$, to the end of X_σ. (See Figure 10(a).) We will also have to consider the case where a string v begins to the right of s and extends beyond the right end of s. We call v an *interloper*. We first consider the case where there are no interlopers, then when there is an interloper on one side, and finally when there is an interloper on each side.

If there are no interlopers, then by the definition of interloper, we only have to extend t left or right to the end of string s. Therefore $\mathrm{E}_\ell(t,\sigma') \leq \mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma)$ and $\mathrm{E}_r(t,\sigma') \leq \mathrm{d}(c_j) - \mathrm{E}_r(s,\sigma)$:

$$
\begin{aligned}
\mathrm{Cost}(t,\sigma') &= |y_{\sigma'}| + \min\{\mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma), \mathrm{d}(c_j) - \mathrm{E}_r(s,\sigma)\} + |\sigma'| \\
&\leq |y_{\sigma'}| + \mathrm{d}(c_j) - \tfrac{1}{2}(\mathrm{E}_\ell(s,\sigma) + \mathrm{E}_r(s,\sigma)) + |\sigma'| \\
&= \tfrac{1}{2}|y_{\sigma'}| + \tfrac{1}{2}|y_\sigma| + |\sigma'| && \text{(Eq. 8.)} \\
&< \mathrm{d}(c_j) + \tfrac{1}{2}|\sigma| + \tfrac{3}{2}|\sigma'| && \text{(Eq. 5.)} \\
&< \tfrac{3}{2}(\mathrm{d}(c_j) + |\sigma'|).
\end{aligned}
$$

Suppose there is an interloper on one side. Let v be the interloper that extends the furthest to the right as in Figure 10(b). Because all strings must have a $\frac{2}{3}$-repeater, let σ_2 be the smallest $\frac{2}{3}$-repeater of v. By our conditions on where v starts and ends, X_{σ_2} must be linked with X_σ and contain $X_{\sigma'}$ as shown. We know by our choice of s and σ that $|\sigma| > |\sigma_2|$. By Lemma 4.6, $|\sigma_2| > \frac{1}{2}\mathrm{d}(c_j)$, so we apply Lemma 4.8 to conclude that

$$|y_\sigma| < \frac{3}{2}\mathrm{d}(c_j). \tag{9}$$

If v goes beyond X_σ to the right as in the Figure, we will extend t to the left. As above when there were no interlopers, we use $\mathrm{E}_\ell(t,\sigma') = \mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma)$:

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &= |y_{\sigma'}| + \mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma) + |\sigma'| \\
&= |y_{\sigma'}| + \mathrm{d}(c_j) - (|y_{\sigma'}| + 2\mathrm{d}(c_j) - |y_\sigma| - \mathrm{E}_r(s,\sigma)) + |\sigma'| && \text{(Eq. 8.)} \\
&= |y_\sigma| + \mathrm{E}_r(s,\sigma) - \mathrm{d}(c_j) + |\sigma'| \\
&< |y_\sigma| + |\sigma'| && (\mathrm{E}_\ell(s,\sigma) < \mathrm{d}(c_j).) \\
&< \tfrac{3}{2}(\mathrm{d}(c_j) + |\sigma'|). && \text{(Eq. 9.)}
\end{aligned}$$

Finally, suppose that there is an interloper in each direction, say w and v with $\frac{2}{3}$-repeaters σ_1 and σ_2 respectively, as in Figure 10(c). Although this seems to present some difficulties, the situation also gives us stronger bounds because multiple characteristics are linked and we can employ Lemma 4.8.

Note that X_{σ_1} and X_{σ_2} are linked, as are X_{σ_1} and X_σ. We derive a lower bound on $|\sigma|$:

$$\begin{aligned}
|\sigma| &> |\sigma_1| + |y_\sigma| - \mathrm{d}(c_j) && \text{(Lemma 4.8.)} \\
&> |\sigma_2| + |y_{\sigma_1}| - \mathrm{d}(c_j) + |y_\sigma| - \mathrm{d}(c_j) && \text{(Lemma 4.8.)} \\
&> |y_{\sigma_1}| + |y_\sigma| - \tfrac{3}{2}\mathrm{d}(c_j) && \text{(Lemma 4.6.)} \\
\Rightarrow \tfrac{1}{3}|\sigma| &> \tfrac{2}{3}|\sigma_1| - \tfrac{1}{6}\mathrm{d}(c_j) && \text{(Def. 3.2.)} \\
\Rightarrow |\sigma_1| &< \tfrac{1}{4}\mathrm{d}(c_j) + \tfrac{1}{2}|\sigma|.
\end{aligned}$$

Without loss of generality let $|\sigma_1| > |\sigma_2|$. We will choose to extend in the direction of the larger of σ_1 and σ_2, so in this case we will extend t to the left. Because X_{σ_1} and X_σ are linked, we conclude that

$$\mathrm{d}_{\mathrm{pref}}(y_{\sigma_1}, y_\sigma) < |y_{\sigma_1}| - \mathrm{d}(c_j). \tag{10}$$

We use Equation 10, Lemma 4.8, and Equation 4.3 to bound g_1:

$$\begin{aligned}
\mathrm{d}_{\mathrm{pref}}(y_{\sigma_1}, y_\sigma) &< |y_{\sigma_1}| - \mathrm{d}(c_j) \\
&< |\sigma_1| - |\sigma_2| \\
&< \frac{1}{4}\mathrm{d}(c_j) + \frac{1}{2}|\sigma| - |\sigma_2| \\
&< \frac{1}{2}|\sigma| - \frac{1}{4}\mathrm{d}(c_j).
\end{aligned}$$

We now calculate the anticipated cost of extending t to the left (in the direction of σ_1, the larger of σ_1 and σ_2):

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &\leq |y_{\sigma'}| + \mathrm{E}_\ell(t,\sigma') + |\sigma'| \\
&\leq |y_{\sigma'}| + \mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma) + g_1 + |\sigma'| \\
&< |y_{\sigma'}| + \mathrm{d}(c_j) - (\tfrac{5}{3}\mathrm{d}(c_j) + \tfrac{2}{3}|\sigma| - |y_\sigma|) + g_1 + |\sigma'| && \text{(Case bound.)} \\
&= |y_{\sigma'}| - \tfrac{11}{12}\mathrm{d}(c_j) - \tfrac{1}{6}|\sigma| + |y_\sigma| + |\sigma'| && \text{(Eq. 11.)} \\
&< |y_{\sigma'}| - \tfrac{11}{12}\mathrm{d}(c_j) - \tfrac{1}{6}|\sigma| + (|\sigma| - |\sigma_1| + \mathrm{d}(c_j)) + |\sigma'| && \text{(Lemma 4.8.)} \\
&= |y_{\sigma'}| + \tfrac{1}{12}\mathrm{d}(c_j) + \tfrac{5}{6}|\sigma| - |\sigma_1| + |\sigma'|.
\end{aligned}$$

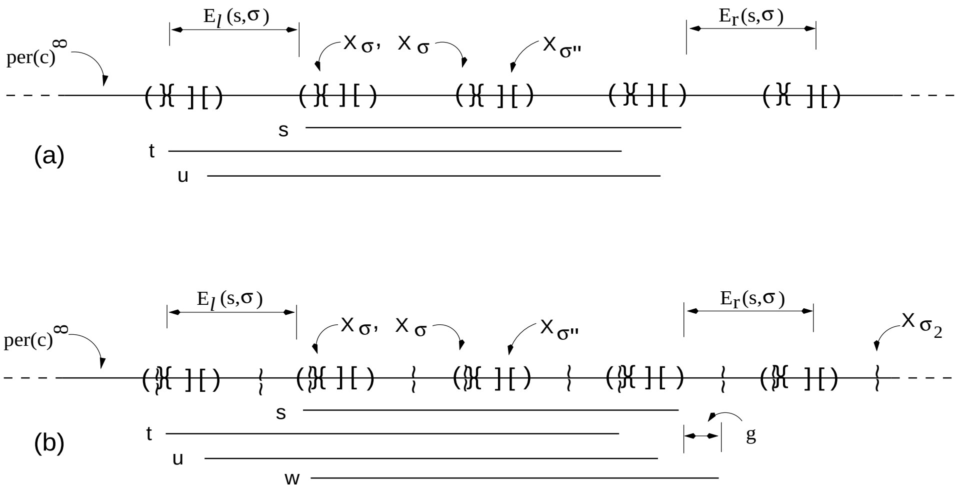


Figure 11: Case 3 of Lemma 4.15. (a) Without an interloper. (b) With an interloper w.

In the last inequality above we were able to apply Lemma 4.8 because X_σ and X_{σ_1} are linked; now we can apply it again, because X_{σ_1} and X_{σ_2} are also linked.

$$
\begin{aligned}
&< |y_{\sigma'}| + \tfrac{1}{12}\mathrm{d}(c_j) + \tfrac{5}{6}|\sigma| - (|\sigma_2| + |y_{\sigma_1}| - \mathrm{d}(c_j)) + |\sigma'| && \text{(Lemma 4.8.)}\\
&= |y_{\sigma'}| + \tfrac{13}{12}\mathrm{d}(c_j) + \tfrac{5}{6}|\sigma| - |\sigma_2| - |y_{\sigma_1}| + |\sigma'| \\
&< |y_{\sigma'}| + \tfrac{13}{12}\mathrm{d}(c_j) + \tfrac{5}{6}|\sigma| - |\sigma_2| - (\tfrac{2}{3}\mathrm{d}(c_j) + \tfrac{2}{3}|\sigma_1|) + |\sigma'| && \text{(Def. 3.2.)}\\
&< |y_{\sigma'}| + \tfrac{5}{12}\mathrm{d}(c_j) + \tfrac{5}{6}|\sigma| - \tfrac{5}{3}|\sigma_2| + |\sigma'| && (|\sigma_1| > |\sigma_2|.)\\
&< \tfrac{9}{4}\mathrm{d}(c_j) + 2|\sigma'| - \tfrac{5}{3}|\sigma_2| && \text{(Eq. 5.)}\\
&< \tfrac{17}{12}\mathrm{d}(c_j) + 2|\sigma'| \\
&< \tfrac{29}{18}(\mathrm{d}(c_j) + |\sigma'|). && (|\sigma'| < \tfrac{1}{2}\mathrm{d}(c_j).)
\end{aligned}
$$

This concludes the analysis of Case 2A.

In the remaining two cases, $t \neq u$; that is, $\mathrm{LString}(c, \sigma) \neq \mathrm{RString}(c, \sigma)$. Let σ' be the smallest $\frac{2}{3}$-repeater of t and σ'' be the smallest $\frac{2}{3}$-repeater of u, and without loss of generality let $|\sigma'| > |\sigma''|$. By our choice of s we know that $|\sigma| > |\sigma'| > |\sigma''|$.

If $X_{\sigma'}$ is linked with X_σ, we observe that $\mathrm{E}_\ell(s, \sigma) = \mathrm{d}_{\mathrm{pref}}(y', y)$ and $\mathrm{E}_r(s, \sigma) = \mathrm{d}_{\mathrm{suff}}(y'', y)$, so we can apply Lemma 4.11 and conclude that $\min\{\mathrm{E}_\ell(s, \sigma), \mathrm{E}_r(s, \sigma)\} < \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$. This satisfies the bound for Case 1. We therefore only need to consider two remaining cases: when neither $X_{\sigma'}$ nor $X_{\sigma''}$ is linked with X_σ (Case 2B), and when only $X_{\sigma''}$ is linked with X_σ (Case 2C).

Case 2B: $\min\{\mathrm{E}_\ell(s, \sigma), \mathrm{E}_r(s, \sigma)\} > \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$, $X_{\sigma'}$ and $X_{\sigma''}$ both nested.

We show that t can be extended to the right within our bounds. (See Figure 11a.) Here again interlopers are possible, so we will first consider the case without an interloper, and then the case with an interloper on at least one side.

If there is no interloper, then we only have to extend t to the right as far as the end of X_σ. We use Lemma 4.10, the Case bound on $\mathrm{E}_\ell(s, \sigma)$ and $\mathrm{E}_r(s, \sigma)$, and the fact that $\mathrm{E}_\ell(s, \sigma) + \mathrm{E}_r(s, \sigma) =$

$|y_{\sigma'}| + 2\mathrm{d}(c_j) - |y_\sigma|$:

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &\le |y_{\sigma'}| + \mathrm{E}_\ell(t,\sigma') + |\sigma'| \\
&\le |y_{\sigma'}| + \mathrm{d}_{\mathrm{suff}}(y',y) + |\sigma'| \\
&= |y_{\sigma'}| + (|y_\sigma| - |y_{\sigma'}| - \mathrm{d}_{\mathrm{pref}}(y',y)) + |\sigma'| \\
&= |y_\sigma| + \mathrm{E}_\ell(s,\sigma) - \mathrm{d}(c_j) + |\sigma'| \\
&= \tfrac{5}{3}|\sigma'| + |y_\sigma| + \mathrm{E}_\ell(s,\sigma) - \mathrm{d}(c_j) - \tfrac{2}{3}|\sigma'|.
\end{aligned}$$

We apply Lemma 4.10 and the fact that $\mathrm{E}_\ell(s,\sigma) = \mathrm{d}_{\mathrm{pref}}(y',y)$ to bound the last term above,

$$\begin{aligned}
&\le \tfrac{5}{3}|\sigma'| + |y_\sigma| + \mathrm{E}_\ell(s,\sigma) - \mathrm{d}(c_j) - \tfrac{2}{3}(\mathrm{E}_\ell(s,\sigma) + |y_\sigma| - 2\mathrm{d}(c_j)) \\
&= \tfrac{5}{3}|\sigma'| + \tfrac{1}{3}|y_\sigma| + \tfrac{1}{3}\mathrm{E}_\ell(s,\sigma) + \tfrac{1}{3}\mathrm{d}(c_j) \\
&< \tfrac{5}{3}|\sigma'| + \tfrac{4}{3}\mathrm{d}(c_j) && \text{(Eq. 5.)} \\
&< \tfrac{5}{3}(\mathrm{d}(c_j) + |\sigma'|).
\end{aligned}$$

Because $|\sigma'| > |\sigma''|$ and σ' is a $\frac{2}{3}$-repeater of t, t was our choice of representative and we elected to extend to the right. Therefore the only interloper which concerns us is one like w in Figure 11b. Let σ_2 be the smallest $\frac{2}{3}$-repeater of w. Due to our choice of s, $|\sigma| > |\sigma_2|$, and we can apply Lemma 4.8 to obtain:

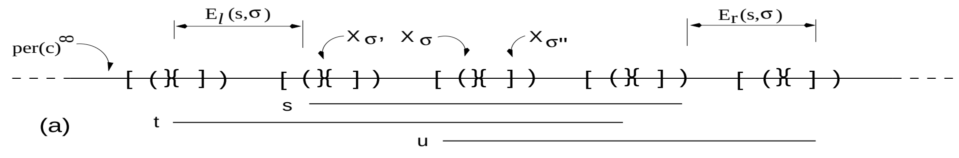
$$|\sigma_2| < |\sigma| - |y_\sigma| + \mathrm{d}(c_j). \tag{11}$$

We bound the distance that the interloper w extends beyond X_σ:

$$\begin{aligned}
\mathrm{d}_{\mathrm{suff}}(y, y_{\sigma_2}) &< |y_{\sigma_2}| - \mathrm{d}(c_j) - (\mathrm{E}_\ell(s,\sigma) - 2\mathrm{d}(c_j) - |y_\sigma|) \\
&= |y_{\sigma_2}| + \mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma) - |y_\sigma|.
\end{aligned}$$

We now calculate the cost of extending t to the right:

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &\le |y_{\sigma'}| + \mathrm{E}_r(t,\sigma') + |\sigma'| \\
&\le |y_{\sigma'}| + \mathrm{d}_{\mathrm{suff}}(y_{\sigma'}, y_{\sigma_2}) + g + |\sigma'| \\
&= |y_{\sigma'}| + (|y_\sigma| - (\mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma)) - |y_{\sigma'}|) + g + |\sigma'| \\
&= |y_\sigma| - \mathrm{d}(c_j) + \mathrm{E}_\ell(s,\sigma) + g + |\sigma'| \\
&< |y_\sigma| - \mathrm{d}(c_j) + \mathrm{E}_\ell(s,\sigma) \\
&\quad + (|y_{\sigma_2}| + \mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma) - |y_\sigma|) + |\sigma'| && \text{(Eq. 12.)} \\
&= |y_{\sigma_2}| + |\sigma'| \\
&< \mathrm{d}(c_j) + |\sigma_2| + |\sigma'| && \text{(Eq. 5.)} \\
&< \mathrm{d}(c_j) + |\sigma'| + (|\sigma| - |y_\sigma| + \mathrm{d}(c_j)) && \text{(Eq. 11.)} \\
&< \mathrm{d}(c_j) + |\sigma'| + \tfrac{1}{3}|\sigma| + \tfrac{1}{3}\mathrm{d}(c_j) && \text{(Def. 3.2.)} \\
&< \tfrac{5}{3}(\mathrm{d}(c_j) + |\sigma'|).
\end{aligned}$$



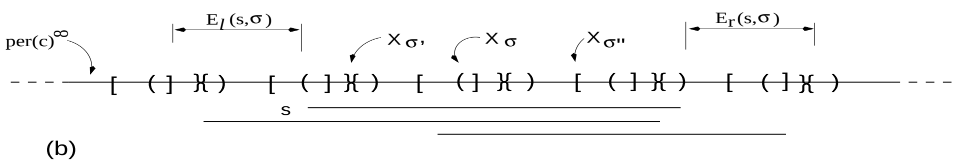


Figure 12: Case 4 of Lemma 4.15. (a) $X_{\sigma'}$ nested within $X_{\sigma''}$. (b) $X_{\sigma'}$ not nested within $X_{\sigma''}$.

Case 4: $\min\{\mathrm{E}_\ell(s,\sigma), \mathrm{E}_r(s,\sigma)\} > \frac{5}{3}\mathrm{d}(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|$, $X_{\sigma''}$ (but not $X_{\sigma'}$) linked with X_σ.
In this case $X_{\sigma'}$ might be nested within $X_{\sigma''}$ (Figure 12a), or not (Figure 12b). It is an unlikely case to give us trouble, because here the smaller $\frac{2}{3}$-repeater has the larger characteristic. In fact, it turns out that we achieve a stronger bound here than in other cases.
Subcase (i). Because $X_{\sigma''}$ contains $X_{\sigma'}$, Lemma 4.6 applies, so $|\sigma''| > \frac{1}{2}\mathrm{d}(c_j)$. Since Lemma 4.8 also applies we have

$$|y_\sigma| < |\sigma| - |\sigma''| + \mathrm{d}(c_j) < \frac{3}{2}\mathrm{d}(c_j). \tag{12}$$

If there are no interlopers, we can now bound the anticipated cost of extending t to the right as follows:

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &\le |y_{\sigma'}| + \mathrm{E}_\ell(t,\sigma') + |\sigma'| \\
&\le |y_{\sigma'}| + \mathrm{d}_{\mathrm{suff}}(y_{\sigma'}, y_\sigma) + |\sigma'| \\
&= |y_{\sigma'}| + (|y_\sigma| - |y_{\sigma'}| - (\mathrm{d}(c_j) - \mathrm{E}_\ell(s,\sigma))) + |\sigma'| \\
&< |y_\sigma| + |\sigma'| && (\mathrm{E}_\ell(s,\sigma) < \mathrm{d}(c_j)) \\
&< \tfrac{3}{2}(\mathrm{d}(c_j) + |\sigma'|). && (\text{Eq. } 12)
\end{aligned}$$

Now suppose there was an interloper v with smallest $\frac{2}{3}$-repeater σ_2. Then X_{σ_2} would be linked with $X_{\sigma''}$ and X_σ, and Lemma 4.11 would apply as in Figure 7(d), and we would once again be in Case 1.
Subcase (ii). Now $X_{\sigma'}$ is not nested within $X_{\sigma''}$, as in Figure 12b. If there are no interlopers, then we only have to extend t to the right to the end of X_σ:

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &\le |y_{\sigma'}| + \mathrm{E}_\ell(t,\sigma') + |\sigma'| \\
&\le |y_{\sigma'}| + \mathrm{E}_\ell(s,\sigma) + \mathrm{d}(c_j) - |y_{\sigma'}| - (2\mathrm{d}(c_j) - |y_\sigma|) + |\sigma'| \\
&= |y_\sigma| + \mathrm{E}_\ell(s,\sigma) - \mathrm{d}(c_j) + |\sigma'|.
\end{aligned}$$

We apply Lemma 4.10 to complete the analysis:

$$\begin{aligned}
\mathrm{Cost}(t,\sigma') &< |y_\sigma| + (|\sigma'| - |y_\sigma| + 2\mathrm{d}(c_j)) - \mathrm{d}(c_j) + |\sigma'| \\
&= 2|\sigma'| + \mathrm{d}(c_j) \\
&< \frac{3}{2}(\mathrm{d}(c_j) + |\sigma'|.
\end{aligned}$$

As in Case (i), if there is an interloper then Lemma 4.11 will apply (Figures 7c or d), and we have Case 1.

This completes the proof of Case 4, which completes the proof of the lemma. ■

We now combine Lemmas 2.2, 4.12, 4.13, and 4.15 to obtain:

Theorem 4.16 *Algorithm* G-SHORTSTRING *is a* $2\frac{2}{3}$*-approximation algorithm for the shortest superstring problem.*

5 Discussion

There remains a large gap between the best known approximation algorithms for this problem and the lower bound of $1+\epsilon$ for some very small constant ϵ. There is also a gap between the best proven bound on the performance of a simple greedy algorithm for the problem [3], a 4-approximation, and the lower bound of 2 for that algorithm. This gap is of particular interest because the greedy algorithm is simple and fast and therefore is used in practice for DNA sequencing. We believe that the techniques described in this paper, as well as those being developed by other researchers, will contribute to closing these gaps.

References

[1] C. Armen and C.Stein. Improved length bounds for the shortest superstring problem. In *Proceedings of Workshop on Algorithms and Data Structures*, pages 494–505, 1995.

[2] C. Armen and C. Stein. Short superstrings and the structure of overlapping strings. *J. of Computational Biology*, 2(2):307–332, 1995.

[3] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. *Journal of the ACM*, 41(4):630–647, July 1994.

[4] Dany Breslauer, Tao Jiang, and Zhigen Jiang. Rotations of periodic strings and short superstrings. *Journal of Algorithms*, 24(2):340–353, August 1997.

[5] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*. MIT Press/McGraw-Hill, 1990.

[6] A. Czumaj, L. Gasieniec, M. Piotrow, and W. Rytter. Parallel and sequential approximations of shortest superstrings. In *Proceedings of Fourth Scandinavian Workshop on Algorithm Theory*, pages 95–106, 1994.

[7] A. Lesk (edited). *Computational Molecular Biology, Sources and Methods for Sequence Analysis.* Oxford University Press, 1988.

[8] N. Fine and H. Wilf. Uniqueness theorems for periodic functions. *Proceedings of the American Mathematical Society*, 16:109–114, 1965.

[9] A.M. Frieze, G. Galbiati, and F. Maffoli. On the worst case performance of some algorithms for the asymmetric travelling salesman problem. *Networks*, 12:23–39, 1982.

[10] J. Gallant, D. Maier, and J. Storer. On finding minimal length superstrings. *Journal of Computer and System Sciences*, 20:50–58, 1980.

[11] D. Gusfield, G. Landau, and B. Schieber. An efficient algorithm for the all pairs suffix-prefix problem. *Information Processing Letters*, (41):181–185, March 1992.

[12] Tao Jiang and Ming Li. Approximating shortest superstrings with constraints. *Theoretical Computer Science*, (134):473–491, 1994.

[13] J.D. Kececioglu and E.W. Myers. Combinatorial algorithms for dna sequence assembly. *Algorithmica*, 13(1/2):7–51, 1995.

[14] John D. Kececioglu. *Exact and approximation algorithms for DNA sequence reconstruction.* PhD thesis, University of Arizona, 1991.

[15] R. Kosaraju, J. Park, and C. Stein. Long tours and short superstrings. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, November 1994.

[16] M. Li. Towards a DNA sequencing theory (learning a string). In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 125–134, 1990.

[17] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization, Algorithms and Complexity.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[18] H. Peltola, H. Soderlund, J. Tarjio, and E. Ukkonen. Algorithms for some string matching problems arising in molecular genetics. In *Proceedings of the IFIP Congress*, pages 53–64, 1983.

[19] Graham A. Stephen. *String searching algorithms.* World Scientific, 1994.

[20] J. Storer. *Data compression: methods and theory.* Computer Science Press, 1988.

[21] Z Sweedyk. A 2-1/2 approximation algorithm for shortest superstring. *SIAM Journal on Computing.* To appear.

[22] Shang-Hua Teng and Frances Yao. Approximating shortest superstrings. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 158–165, November 1993.

[23] J. Turner. Approximation algorithms for the shortest common superstring problem. *Information and Computation*, 83:1–20, 1989.