A 2\(2/3\) superstring approximation algorithm

Chris Armen\(^a,^*\), Clifford Stein\(^b,^1\)

\(^a\) Department of Computer Science, University of Hartford, W. Hartford, CT 06117-1599, USA
\(^b\) Dartmouth College, Hanover, NH 03755-3510, USA

Received 21 February 1997; received in revised form 18 March 1998; accepted 15 April 1998

Abstract

Given a collection of strings \(S = \{s_1, \ldots, s_n\}\) over an alphabet \(\Sigma\), a superstring \(x\) of \(S\) is a string containing each \(s_i\) as a substring; that is, for each \(i, 1 \leq i \leq n\), \(x\) contains a block of \(|s_i|\) consecutive characters that match \(s_i\) exactly. The shortest superstring problem is the problem of finding a superstring \(x\) of minimum length. The shortest superstring problem has applications in both data compression and computational biology. It was shown by Blum et al. (1994) to be MAX SNP-hard. The first \(O(1)\)-approximation algorithm also appeared in Blum et al. (1994), 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 \(2\frac{1}{4}\)-approximation Armen and Stein (1995). We present our new algorithm, G-SHORTSTRING, which achieves an approximation ratio of \(2\frac{2}{3}\). Our approach builds on the work in Armen and Stein (1995) 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. 1998 Published by Elsevier Science B.V. All rights reserved.

1. Introduction

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

* Corresponding author. E-mail: armen@hartford.edu.

\(^1\) Research partly supported by NSF Award CCR-9308701, a Walter Burke Research Initiation Award and a Dartmouth College Research Initiation Award.

0166-218X/98/$19.00 1998 Published by Elsevier Science B.V. All rights reserved.

PII S0166-218X(98)00065-1
The shortest superstring problem has applications in both computational biology [8, 17, 19] and data compression [11, 21]. 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.

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 et al. [11]; it was later shown to be MAX SNP-hard [4]. The first $O(1)$-approximations were given by Blum et al. [4], 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 $2.8^5$, $2.5^5$, and slightly better than 2.8 (see [7, 16, 23], respectively). Our result of $2.3^3$ [2, 3] was the best known until recently, and when combined with the algorithm of [16] achieves an approximation ratio of about 2.725.

In this paper we describe our $2.3^3$-approximation algorithm for the shortest superstring problem. Algorithmically, our approach is a generalization of the one taken in [2], 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 $|u| - ov(u, v)$, where $ov(u, v)$ 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 [4] uses a theorem about infinite periodic functions [9], 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. Independent of our work,
Breslauer et al. [5] obtained a 2.596-approximation by proving other structural character-
istics of periodic strings. More recently, Sweedyk [22] 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 [4] 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 - \varepsilon)-approximation, or a lower bound result.

2. Preliminaries

For consistency, we use some notation and definitions of [4, 23]. We introduce this
notation rather tersely; the reader interested in a more detailed introduction is referred
to [4] or [1]. 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 \( \text{opt}(S) \) to denote the length
of the shortest superstring of \( S \).

Given two strings \( s \) and \( t \), we define \( \text{ov}(s, t) \), 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 \( t \), \( \text{pref}(s, t) \), and refer to \(|u|\) as the prefix distance from \( s \) to \( t \), \( d_{\text{pref}}(s, t) \).
Observe that for any \( s \) and \( t \), \( \text{ov}(s, t) + d_{\text{pref}}(s, t) = |s| \). String \( u xv \), the shortest super-
string of \( s \) and \( t \) in which \( s \) appears before \( t \) is denoted by \( (s, t) \), and \(|(s, t)| = |s| + |t|\)
\( - \text{ov}(s, t) \).

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 \( d_{\text{pref}}(s_i, s_j) \)
and label the edge with \( \text{pref}(s_i, s_j) \). We can now observe that a minimum length
hamiltonian cycle (traveling salesman tour) \( v_{\pi_1}, \ldots, v_{\pi_k}, v_{\pi_1} \), in \( G \), with edge \((i, j)\) labeled
by \( \text{pref}(s_i, s_j) \), almost corresponds to a superstring in \( S \), the only difference being that
we must replace \( \text{pref}(s_{\pi_k}, s_{\pi_1}) \) with \( s_{\pi_k} \). Since \(|\text{pref}(s_i, s_j)| \leq |s_i|\), we can conclude that
\( \text{opt}(\text{TSP}) \leq \text{opt}(S) \), where \( \text{opt}(\text{TSP}) \) is the optimal solution to TSP defined above. This
TSP is directed (sometimes called asymmetric); thus the best known approximation
[10] 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 $d(C)$ 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 [18]. Since a tour is a cycle cover, $d(C) \leq \text{opt(TSP)}$. As noted above, $\text{opt(TSP)} \leq \text{opt(S)}$, so the weight of the cycle cover $d(C)$ gives us a lower bound on the length of the optimal solution $\text{opt(S)}$.

Because $d_{\text{pref}}(s_i, s_j) + \text{ov}(s_i, s_j) = |s_i|$, 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. [24, 23, 16]) 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 concatenation of $k$ copies of a string $t$, and $t^\infty$ 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 $\text{per}(c)$ 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 $\text{per}(c)$, 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 $\text{per}(c)^\infty$.

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

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 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:**

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

$$|\langle s_1, c \rangle| = d(c) + |s_1|.$$

(1)
Right extension:

\[ \langle s_1, c \rangle^+ = \langle s_1, s_2, \ldots, s_m \rangle = \text{pref}(s_1, s_2) \cdot \text{pref}(s_2, s_3) \cdot \ldots \cdot \text{pref}(s_{m-1}, s_m) \cdot s_m, \]

\[ |\langle s_1, c \rangle^+| = d(c) + \text{ov}(s_m, s_1). \quad (2) \]

Left extension:

\[ \langle s_1, c \rangle^- = \langle s_2, s_3, \ldots, s_m, s_1 \rangle = \text{pref}(s_2, s_3) \cdot \text{pref}(s_3, s_4) \cdot \ldots \cdot \text{pref}(s_m, s_1) \cdot s_1, \]

\[ |\langle s_1, c \rangle^-| = d(c) + \text{ov}(s_1, s_2). \quad (3) \]

The three types of extension are illustrated in Fig. 1 for string \( s_1 \). The definitions of \( \langle s_1, c \rangle \), \( \langle s_1, c \rangle^+ \), and \( \langle s_1, c \rangle^- \), \( 2 \leq i \leq m \), are analogous. Notice that \( \langle s_1, c \rangle^+ = \langle s_1, c \rangle^- \); this apparent redundancy will be useful later. Full extension, right extension and left extension form each form superstrings that include all of the strings in \( c \), as proved in [4]. 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 \( \langle s_1, c \rangle \).

We now describe a generic version of a superstring algorithm from [4] that is also used in some form by [3, 7, 23]. An execution of the algorithm appears as in Fig. 2.

**GENERIC SUPERSTRING ALGORITHM**

(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 \( y \in CC \)
   Let \((w_i, x_i)\) be the minimum-overlap edge in \( y \).
   Let \( z_i = (x_i, y_i) \).

(5) For each cycle \( y \in CC \)
   For each representative \( r_c \) in \( y \),
   Replace \( r_c \) with \( \langle r_c, c \rangle \) in \( z_i \).

(6) Concatenate the \( z_i \) to form a superstring \( \alpha \).

As shown in [4], \( \alpha \) 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) and (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 \( y_i \). 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 [4] 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 [3]. For a cycle \( \gamma \in CC \), let \( ov_{\text{cost}}(\gamma) \) be the overlap of the edge deleted in Step (4); this represents the cost of breaking the \( CC \) cycle \( \gamma \) to form a superstring of the representatives in \( \gamma \). Let \( \text{Ext}(\gamma) \) be the length added in Step (6) to extend each representative in \( \gamma \) to include the rest of the strings in its cycle. For GENERIC, \( \text{Ext}(\gamma) \) 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

$$|x| \leq \text{opt}(S) + \sum_{\gamma \in CC} (\text{ovcost}(\gamma) + \text{Ext}(\gamma)).$$

The following key lemma from [4] is essential to this analysis:

Lemma 2.1 (Blum et al. [4]). 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 $\sum_{\gamma \in CC} \text{ovcost}(\gamma)$ in (4) is less than $d(C)$, because it provides a bound on any overlap edge in terms of $d(C)$. The contribution of $\sum_{\gamma \in CC} \text{Ext}(\gamma)$, the second term, is exactly $d(C)$. Because the TSP is a cycle cover, $d(C) \leq \text{opt}(S)$, 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 $\gamma$ has three or more vertices, then $\text{ovcost}(\gamma) \leq \frac{2}{3} \sum_{c \in \gamma} d(c)$. 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 $\gamma$ in $CC$ consisting of vertices $v$ and $t$, attains a bound of $\text{ovcost}(\gamma) + \text{Ext}(\gamma) \leq \beta(d(c_v) + d(c_t))$, for some $\beta \geq \frac{3}{2}$, 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 [4] 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 $t^\infty$. Note that for a cycle $c$ in a minimum cycle cover, $\text{per}(c)$ 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 (Blum et al. [4]). Let $w$ be a substring of both $(\sigma_j)^\infty$ and $(\sigma_k)^\infty$ for two strings $\sigma_j$ and $\sigma_k$. Then if $|w| \geq |\sigma_j| + |\sigma_k|$, at least one of $\sigma_j$ or $\sigma_k$ 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 [3].

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 \( \gamma \) in \( CC \) in which one of \( ov(r_j, r_k) \) or \( ov(r_k, r_j) \) is large but the other is small. In Step (4) we will break \( \gamma \) to form a string, by choosing \( \langle r_j, \gamma \rangle + \) or \( \langle r_k, \gamma \rangle - \); 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 \( \gamma \) be a 2-cycle in the second cycle cover \( CC \) of the \textsc{Generic} algorithm, consisting of vertices \( r_j \) and \( r_k \), the representatives of cycles \( c_j \) and \( c_k \) in \( C \). Then \( \gamma \) is a \( g\text{-HO2-cycle} \) if \( \min\{ov(r_j, r_k), ov(r_k, r_j)\} \geq g(d(c_j) + d(c_k)) \) for some real \( g > 0 \).

In this paper we will be interested in \( \frac{2}{3}\text{-HO2-cycles} \). Our strategy is to anticipate, when we choose representatives, the potential of each string to participate in a \( \frac{2}{3}\text{-HO2-cycle} \). In particular, 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 \( \sigma \). In order for the high-overlap 2-cycle to occur, this smaller period \( \sigma \) must correspond to the period of another representative \( w \) such that \( ov(w, z) \) would be large. Similarly, the suffix of \( z \) must have the same smaller period, so that \( ov(z, w) \) would be large. We require some notation to describe this potential.

**Definition 3.2.** Let \( z \) be a string in cycle \( c \) and let \( \sigma \) be an irreducible string with \( |\sigma| < d(c) \). Then \( \sigma \) is a \( g\text{-repeater} \) of \( z \) if there exist witness strings \( y_\ell \) and \( y_r \), such that

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

We will always choose \( y_\ell \) and \( y_r \) to be the maximum length prefix and suffix that satisfy conditions 1–3 above.

---

This is a less general definition than that used in [2]. The original definition had a second parameter, which allows for different weights for \( d(c) \) and \( |\sigma| \) in condition 3 of the definition. This flexibility is not required for the present result.
In this paper we will always be concerned with $^2_3$-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 $\sigma = ababad$, $\text{per}(c) = ababadababadabab$, $y_f = ababadababadababa$ and $y_r = ababadababadabab$. So $|y_f|, |y_r| > \frac{2}{3}(d(c) + |\sigma|)$, and we say that $\sigma$ is a $^2_3$-repeater of $z$.

We can for simplicity define one witness $y_\sigma$ which contains both $y_f$ and $y_r$; that is, we define $y_\sigma$ 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 $\sigma$ and try to repeat it as many times as possible, in both directions, while being consistent with $c$, you get $y_\sigma$. Henceforth, when discussing and proving properties of cycles, we will refer to the unique maximal witness $y_\sigma$ rather than to the underlying pair of witnesses $y_f$ and $y_r$. This simplification is conservative. When the context is clear, we will drop the $\sigma$ and just refer to the witness $y$.

A witness $y$ for some $^2_3$-repeater $\sigma$ satisfies

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

(5)

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

A copy of $y$ occurs once every $d(c)$ in $\text{per}(c)^\infty$. A copy of each string in $c$ also occurs every $d(c)$. Recall that we defined $^2_3$-repeaters (Definition 3.2) in terms of some string $z$ in a cycle $c$ that contains witnesses $y_f$ 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 $^2_3$-repeater $\sigma$ if $\sigma$ is a $^2_3$-repeater of any string in $c$.

We can now present our algorithm; additional notation and properties of $^2_3$-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 $r$, the possible involvement of $r$ 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 $\text{per}(c)\infty$, for instance between two witness strings $y$ and $y'$. As noted above, a copy of any substring of $\text{per}(c)\infty$ occurs every $d(c)$ in $\text{per}(c)\infty$. We overload our notation for $d_{\text{pref}}(\cdot,\cdot)$ and $\text{ov}(\cdot,\cdot)$ in the obvious way to refer to prefix distance $d_{\text{pref}}(y, y')$ and overlap $\text{ov}(y, y')$. That is, if we fix any copy of $y$ in $\text{per}(c)\infty$, $d_{\text{pref}}(y, y')$ is the distance from the beginning of $y$ to the beginning of the next copy of $y'$, and $\text{ov}(y, y')$ is the overlap between the same copies. In addition, we define $\text{ov}(y, y')$ 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 the original set of strings $S$, which is substring-free.) We also define the suffix distance $d_{\text{suff}}(y, y')$ 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); i.e., it concatenates each representative $r_c$ with a copy of $\text{per}(c)$. This adds length $d(c)$ 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 $\text{ov}(t, v)$ or $\text{ov}(v, t)$, but not both. Recall that $\text{Ext}(\gamma)$ is the length added by the algorithm to include the remaining strings in the cycles represented by the strings in $\gamma$. For any 2-cycle $\gamma$, our algorithm will extend only as far as is necessary to include the remaining strings in each representative's cycle, thus reducing $\text{Ext}(\gamma)$ to less than the sum of the weights of the cycles in $\gamma$. The following definitions formalize this idea (Fig. 4).

**Definition 4.1.** Let $\sigma$ be a $\frac{2}{3}$-repeater with maximal witness $y_\sigma$ in an $m$-vertex cycle $c$. Index the strings $s_1, s_2, \ldots, s_m$ so that $d_{\text{pref}}(y_\sigma, s_i) < d_{\text{pref}}(y_\sigma, s_{i+1})$, $1 \leq i < m$. Then we define the right string of $\sigma$ in $c$, $\text{RString}(c, \sigma) = s_m$. The left string of $\sigma$ in $c$, $\text{LString}(c, \sigma) = s_1$. If $\text{Ext}(\gamma)$ is the length added by the algorithm to include the remaining strings in the cycles represented by the strings in $\gamma$. For any 2-cycle $\gamma$, our algorithm will extend only as far as is necessary to include the remaining strings in each representative's cycle, thus reducing $\text{Ext}(\gamma)$ to less than the sum of the weights of the cycles in $\gamma$. The following definitions formalize this idea (Fig. 4).
LString(c, σ) is defined symmetrically; reindex the strings s_i such that d_suff(s_i, y_σ) > d_suff(s_{i+1}, y_σ), 1 ≤ i < m. Then we define LString(c, σ) = 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 RString(c, σ). The idea is that if we choose a string t as representative of cycle c, 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 per(c_v) = σ, then RString(c, σ) is the rightmost string that we will have to include if we extend t to the right. Fig. 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, \( E_r(t, σ) = d_{suff}(y_σ, RString(c, σ)) \). The left σ-extension with respect to t, \( E_l(t, σ) = d_{pref}(LString(c, σ), y_σ) \).

For each string t in cycle c_r, we are interested in the worst-case cost of choosing t as the representative of c_r. 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 y with some string v such that per(c_v) = σ. By Lemma 2.2 we need to bound \( ov_{cost}(γ) + Ext(c_t) + Ext(c_v) \). 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_v be the suffix of t that is the witness string for σ. From the end of y_v, we need to extend to the right only as far as necessary to include RString(c, σ). We also have to extend v to include the remaining strings in c_v; we assume the cost of full extension, which is \( d(c_v) \). 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

\[
Cost(t, σ) = |y_σ| + \min\left\{E_r(t, σ), E_l(t, σ)\right\} + |σ|.
\]

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 Cost(t, σ). We imagine that once we choose the representative r of cycle c, an adversary will match r with another
representative v whose period \( \text{per}(c_v) \) matches some \( \frac{2}{3} \)-repeater in r to form a high-overlap 2-cycle. Allowing \( a \in t \) to mean "a \( \frac{2}{3} \)-repeater of t", we seek

\[
\text{BestRep}(c) = \arg\min_{t \in c} \left\{ \max_{\sigma \in I} \{\text{Cost}(t, \sigma)\} \right\}.
\]

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

**Procedure G-FINDREPS(c)**

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** \( r_j = \text{BestRep}(c_j) \);
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 (5) 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 \( \gamma_i \) in \( CC \):
   - if \( \gamma_i \) is a \( \frac{2}{3} \)-HO2-cycle \((v, t)\)
     - **then if** \( \text{ov}(v, t) + E_r(t, \text{per}(c_v)) \leq \text{ov}(v, t) + E_r(t, \text{per}(c_v)) \)
       - then let \( z_i = \langle -\langle v, c_v \rangle, \langle t, c_t \rangle \rangle \); 
       - else let \( z_i = \langle -\langle t, c_t \rangle, \langle v, c_v \rangle \rangle \);
     - **else** let \( (w_i, x_j) \) be the least-overlap edge in \( \gamma_i \);
       - Let \( z_i = \langle x_i, \gamma_i \rangle \); 
       - For each string \( u \) in \( \gamma_i \)
         - Replace \( u \) with \( \langle u, c_u \rangle \).
4. **Concatenate** the strings \( z_i \) from (4) to form superstring \( \alpha \).

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

---

3 The \textit{argmin} is the argument that achieves the minimum value. For instance, let \( x = 5 \) and \( y = 3 \). Then \( \min(x, y) = 3 \), while \( \arg\min(x, y) = y \).
We are able to prove Lemmas 4.4 and 4.6 using only the size of the $\frac{3}{4}$-repeaters. The remaining technical results in this section require more machinery relating the locations of the $\frac{5}{4}$-repeaters.

In [3] we showed that $\frac{3}{4}$-repeaters are well-parenthesized; i.e., 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 $y = \frac{3}{4}$. We have been able neither to prove this property for $y = \frac{5}{4}$, nor to generate an example of non-well-parenthesized $\frac{3}{4}$-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 Fig. 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 [3], 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** (Armen and Stein [3]). Let $y$ be the maximal witness for a $y$-repeater $\sigma$ 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 $\sigma$ apart are identical, and in $per(c)^\infty$, any two characters that are $d(c)$ 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 $X_\sigma$ 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 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 \( \sigma \) and \( \sigma' \) be two 2-repeaters with positive characteristics in a cycle \( c \), with \( |\sigma| > |\sigma'| \), and \( X_\sigma \) and \( X_{\sigma'} \) linked. Let \( k = \|\sigma||\sigma'|\|. \) Then \( |\sigma| - k|\sigma'| > |y_\sigma| - d(c) \).

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

<table>
<thead>
<tr>
<th>No.</th>
<th>Move</th>
<th>Requirement</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>(+k</td>
<td>\sigma'</td>
<td>)</td>
</tr>
<tr>
<td>2.</td>
<td>(-</td>
<td>\sigma</td>
<td>)</td>
</tr>
<tr>
<td>3.</td>
<td>(-k</td>
<td>\sigma'</td>
<td>)</td>
</tr>
</tbody>
</table>

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

Corollary 4.9. Let \( \sigma \) and \( \sigma' \) be two 2-repeaters with positive characteristics in a cycle \( c \), with \( |\sigma| > |\sigma'| \), and \( X_\sigma \) and \( X_{\sigma'} \) linked. Let \( k = \|\sigma||\sigma'|\|. \) Then \( |\sigma'| > |\sigma| - k|\sigma'| > |y_\sigma| - d(c) \).

Proof. By the choice of \( k \) and Lemma 4.8,

\[ |\sigma'| > |\sigma| - k|\sigma'| > |y_\sigma| - d(c) \]. \( \square \)

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 \( \sigma \), \( \sigma' \) and \( \sigma'' \) 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 \( X_\sigma \). Then \( |\sigma'| > d_{\text{pref}}(y', y) + |y| - 2d(c) \) and \( |\sigma''| > d_{\text{ suff}}(y, y'') + |y| - 2d(c) \).

Proof. Fig. 6 illustrates the start positions of our shift arguments.
\emph{\textsuperscript{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 \( c_r \) and \( c_l \).

The algorithm \textsc{G-ShortString} correctly computes a superstring of the set of strings \( S \). This follows from the correctness of \textsc{Generic}. Our method of choosing representatives for each cycle is a special case of the method of \textsc{Generic}, which chooses an arbitrary string as representative. In step (4b), we do exactly what \textsc{Generic} does. In step (4a), we use a different criterion for breaking a cycle \( \gamma \in \text{CC} \), 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 \( x \).

\textsc{G-ShortString} runs in polynomial time. The distance graph \( G \) can be built in \( O(|S| + n^2) \) time \([12]\), and the cycle cover computations take \( O(n^3) \) time \([18]\). These two results determine the running time of \textsc{Generic}, \( O(|S| + n^3) \). In addition, our procedure \textsc{G-FindReps}(\( c \)) must find all of the \( \frac{2}{3} \)-repeaters in each cycle \( c \in \text{C} \). This can be done naively in polynomial time by examining a prefix and suffix of each string and determining, for each \( j, 2 \leq j < d(c) \), whether the prefix and suffix have periodicity \( j \).

In order to analyze the length bound attained by \textsc{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 \emph{small} \( \frac{2}{3} \)-repeater in a cycle \( c \) is a \( \frac{2}{3} \)-repeater such that \( \frac{2}{3}(d(c) + |\sigma|) < d(c) \), or equivalently if \( |\sigma| < \frac{1}{3} d(c) \). When discussing more than one \( \frac{2}{3} \)-repeater, say \( \sigma \) and \( \sigma' \), we shall often use the relative terms such as \emph{smaller}, \emph{smallest}, \emph{larger}, \emph{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 \( \sigma \) and \( \sigma' \). Let \( y_\gamma(\sigma) \) and \( y_\gamma(\sigma') \) be the prefixes of \( s \) that are the left witness strings of \( \sigma \) and \( \sigma' \) respectively. Let \( y_\gamma = \text{argmin}\{|y_\gamma(\sigma)|, |y_\gamma(\sigma')|\} \) be the prefix of \( s \) which is periodic in both \( \sigma \) and \( \sigma' \). Applying Corollary 2.3, Definition 3.2, and the fact that \( |\sigma'| < \frac{1}{2}d(c) \), we get

\[
|\sigma| > |y_\gamma| - |\sigma'| > \frac{2}{3}d(c) - \frac{1}{3}|\sigma'| > \frac{1}{2}d(c),
\]

a contradiction because \( \sigma \) is a small \( \frac{2}{3} \)-repeater. \( \square \)
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 $\frac{2}{3}$-repeater $\sigma$ and maximal witness $y$. Fix a copy of $y$ in $\per(c)^\infty$. 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 $[-d(c)..d(c)]$. Now mark the point $|y| - d(c)$ and call it the **tail** of $\sigma$. The **characteristic** of $\sigma$, $X_\sigma$, is the interval from the head to the tail. If $|y| - d(c) > 0$ we call $[0..|y| - d(c)]$ a **positive characteristic** $X_\sigma$. If $|y| - d(c) \leq 0$ we call $[|y| - d(c)..0]$ a **negative characteristic** $X_\sigma$.

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 $y_\sigma$ are marked with left and right parentheses; these correspond to the head and tail of adjacent copies of $X_\sigma$.

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 $d(c)$. Intuitively, the characteristic of a repeater borders the portion of $\per(c)$ 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 $X_{\sigma_1}, X_{\sigma_2}$ are **nested** if $X_{\sigma_1}$ is a positive characteristic and $X_{\sigma_2}$ falls within $X_{\sigma_1}$. We say that two characteristics $X_{\sigma_1}, X_{\sigma_2}$ are **disjoint** if their intervals are disjoint. Otherwise we say that $X_\sigma$ 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 $X_\sigma$ be a positive characteristic in cycle $c$ and $X_{\sigma'}$ a characteristic nested within $X_\sigma$ 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

$$|\sigma| + |\sigma'| > \ov(y, y')$$

$$= |y'|$$

$$> \frac{2}{3}(d(c) + |\sigma'|)$$

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

which implies $|\sigma| > \frac{1}{2}d(c)$ because $|\sigma| > |\sigma'|$. \qed
Fig. 6. Proof of Lemma 4.10.

| Start: (A) | Goal: $+|\sigma|$ |
| --- | --- |
| No. | Move | Requirement | Comments |
| 1. | $+|\sigma''|$ | $|\sigma''| \leq d_{\text{suffix}}(y', y'') + |y| - 2d(c)$ | See below |
| 2. | $+|\sigma|$ | $|\sigma| + |\sigma''| < |y| - d(c) + d_{\text{prefix}}(y', y'')$ | See below |
| 3. | $-|\sigma''|$ | $|\sigma| + |\sigma''| > |y| - d(c) + |\sigma''|$ | See below |

| Start: (B) | Goal: $-|\sigma|$ |
| --- | --- |
| No. | Move | Requirement | Comments |
| 1. | $-|\sigma'|$ | $|\sigma'| \leq d_{\text{prefix}}(y', y) + |y| - 2d(c)$ | See below |
| 2. | $-|\sigma|$ | $|\sigma| + |\sigma'| < |y| - d(c) + d_{\text{prefix}}(y', y)$ | See below |
| 3. | $+|\sigma'|$ | $|\sigma| + |\sigma'| > |y| - d(c) + |\sigma'|$ | See below |

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

$$
|\sigma''| > d_{\text{suffix}}(y, y'') + |y| - 2d(c),
$$

$$
|\sigma'| > d_{\text{prefix}}(y', y) + |y| - 2d(c).
$$

Lemma 4.11. Let $\sigma$, $\sigma'$ and $\sigma''$ be $\frac{2}{3}$-repeaters in cycle $c$, with maximal witnesses $y$, $y'$ and $y''$. Let $|\sigma| > |\sigma'| > |\sigma''|$, and let $X_\sigma$ and $X_{\sigma'}$ be positive. Then if $X_{\sigma''}$ is linked with $X_\sigma$, then $min\{d_{\text{prefix}}(y', y), d_{\text{suffix}}(y, y'')\} < \frac{2}{3}d(c) + \frac{2}{3}|\sigma| - |y|.$

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

$$
|\sigma'| + |\sigma''| > d_{\text{prefix}}(y', y) + d_{\text{suffix}}(y, y'') + |y| - 2d(c),
$$

which implies

$$
|\sigma'| > \frac{1}{2}(d_{\text{prefix}}(y', y) + d_{\text{suffix}}(y, y'')) + \frac{1}{2}|y| - d(c).
$$

(6)
We now use Lemma 4.8 and Eq. (6) to obtain
\[ |\sigma| > |\sigma'| + |y| - d(c) \]
\[ > \frac{1}{2}(d_{\text{pref}}(y', y) + d_{\text{suff}}(y, y'')) + \frac{1}{2}|y| - d(c) + |y| - d(c) \]
\[ = \frac{1}{2}(d_{\text{pref}}(y', y) + d_{\text{suff}}(y, y'')) + \frac{3}{2}|y| - 2d(c). \]
Solving for \( \frac{1}{2}(d_{\text{pref}}(y', y) + d_{\text{suff}}(y, y'')) \) and using Definition 3.2 gives us our result:
\[ \frac{1}{2}(d_{\text{pref}}(y', y) + d_{\text{suff}}(y, y'')) < |\sigma| - \frac{3}{2}|y| + 2d(c) \]
\[ < |\sigma| - |y| + 2d(c) - \frac{1}{2}(\frac{3}{2}d(c) + \frac{2}{3}|\sigma|) \]
\[ = \frac{3}{2}d(c) + \frac{2}{3}|\sigma| - |y|. \]

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-\( \frac{3}{3}\)-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 \( \gamma \).

**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 \( \gamma \) for deletion in order to construct a superstring of the representatives in \( \gamma \), as does GENERIC. It then fully extends each
representative \( r \in \gamma \) by the length \( d(c_r) \), as does GENERIC, to include the remaining strings in \( c_r \). \( \square \)

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 \( \gamma \) consisting of representatives \( t \) and \( v \), we need to show that \( \text{ov}_{\text{cost}}(\gamma) + \text{Ext}(\gamma) \leq \frac{2}{3}(d(c_t) + d(c_v)) \). By Definition 4.3, this is equivalent to showing that \( \text{Cost}(t, \text{per}(c_r)) \leq \frac{2}{3}(d(c_t) + d(c_v)) \). When \( \text{Cost}(t) \) was applied in \( \text{BESTREP}(c_t) \), 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 \( \text{per}(c_r) \) for some cycle \( c_r \). 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 \( \gamma \) be a \( \frac{2}{3} \)-HO2-cycle in \( CC \) with \( r_1 \) the representative of cycle \( c_j \) and \( r_k \) the representative of \( c_k \), let \( d(c_j) \neq d(c_k) \), and let all strings in \( c_j \) have at least one small \( \frac{3}{3} \)-repeater. Then \( \text{ov}_{\text{cost}}(\gamma) + \text{Ext}(\gamma) \leq \frac{2}{3}(d(c_j) + d(c_k)) \).

**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 \( d(c_j) \). 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 \( \sigma \) and \( \sigma' \) be small \( \frac{2}{3} \)-repeaters in cycle \( c \). Let \( Q \) be the set of strings that have \( \sigma \) as a \( \frac{2}{3} \)-repeater, and let \( Q' \) be the set of strings that have \( \sigma' \) 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 \( d_{\text{pref}}(t, v) \leq \frac{1}{3}d(c) \); otherwise \( d_{\text{pref}}(v, t) \leq \frac{1}{3}d(c) \) and the same argument follows. Consider the prefixes of \( t \) and \( v \) which are the left witness for \( \sigma \); both prefixes must be substrings of the same copy of \( y_n \). Since \( t' \) is between \( t \) and \( v \), then it also must have a prefix \( y'_n \) which has period \( \sigma \). The same argument holds for the suffixes of \( t \), \( v \) and \( t' \), so \( \sigma \) must be a \( \frac{2}{3} \)-repeater of \( t' \). But then \( t' \) has both \( \sigma \) and \( \sigma' \), contradicting Lemma 4.4. \( \square \)

We resume our analysis. Let \( \sigma_1 \) be the largest of the small \( \frac{2}{3} \)-repeaters in \( c_j \), and let \( Q_1 \) be the set of strings that have \( \sigma_1 \). 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 \( \sigma_i \) 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 \( \ell_i = d_{\text{pref}}(u_{i}, u_{i+1}) \), \( 1 \leq i < j \). (See Fig. 8.)

First we apply Corollary 2.3 to derive a lower bound on \( \ell_1 \):

\[
|\sigma_1| + |\sigma_2| > \alpha v(y_{\sigma_1}, y_{\sigma_2}) \geq |y_{\sigma_1}| - \ell_1,
\]

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

\[
\ell_1 > |y_{\sigma_1}| - 2|\sigma_1|.
\] (7)

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

\[
\alpha v_{\text{cost}}(\gamma) + \text{Ext}(\gamma) = \text{Cost}(u_1, \sigma_1)
\]

\[
= |y_{\sigma_1}| + \min\{E_r(u_1, \sigma_1), E_c(u_1, \sigma_1)\} + |\sigma_1|
\]

\[
\leq |y_{\sigma_1}| + E_r(u_1, \sigma_1) + |\sigma_1|.
\]

If we extend \( u_1 \) to the left, the last string we will have to cover will be \( u_2 \), so \( E_r(u_1, \sigma_1) = d(c) - \ell_1 \), and then we use (7):

\[
\text{Cost}(u_1, \sigma_1) \leq |y_{\sigma_1}| + d(c) - \ell_1 + |\sigma_1|
\]

\[
\leq |y_{\sigma_1}| + d(c) + 3|\sigma_1|
\]

\[
< \frac{5}{3}(d(c) + |\sigma_1|).
\]

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

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

**Lemma 4.15.** Let \( \gamma \) be a \( \frac{3}{2} \)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 there be at least one string in \( c_j \) that does not have a small \( \frac{2}{3} \)-repeater. Then \( \alpha v_{\text{cost}}(\gamma) + \text{Ext}(\gamma) \leq \frac{5}{3}(d(c_j) + d(c_k)) \).
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 \( \text{FINDREPS}(c_j) \).

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 \( \sigma_i \) be the smallest repeater of string \( a_i \). Then let \( s = a_k \), with \( k \) chosen such that \( |\sigma_k| \geq |\sigma_i|, \ 1 \leq i \leq m' \). 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 cases, 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:** \( \min\{E_I(s, \sigma), E_r(s, \sigma)\} < d(c_j) + |y_\sigma| - |y_\sigma| \). If \( E_I(s, \sigma) \leq E_r(s, \sigma) \), then we extend \( s \) to the left; otherwise, we extend \( s \) to the right to cover the remaining strings in \( c \). We bound the cost incurred by \( \gamma \):

\[
\text{ovcost}(\gamma) + \text{Ext}(\gamma) - \text{Cost}(s, \sigma) \\
\leq |y_\sigma| + \min\{E_I(s, \sigma), E_r(s, \sigma)\} + |\sigma| \\
\leq |y_\sigma| + \frac{5}{3}d(c_j) + \frac{2}{3}|\sigma| - |y_\sigma| + |\sigma| \\
= \frac{5}{3}(d(c_j) + |\sigma|).
\]

This concludes the analysis of Case 1.

**Case 2:** \( \min\{E_I(s, \sigma), E_r(s, \sigma)\} > d(c_j) + |y_\sigma| - |y_\sigma| \). If Case 1 does not apply, then, as in Fig. 9, there must be a string \( t = \text{LString}(c_j, \sigma) \) and a string \( u = \text{RString}(c_j, \sigma) \), 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 \( X'_\sigma \) and \( X'_\sigma \) be the copies of \( X_\sigma \) in which \( s \) begins and ends. Then \( t \) must extend into \( X'_\sigma \), because otherwise \( E_I(s, \sigma) \leq 2d(c_j) - |y_\sigma| \leq \frac{5}{3}d(c_j) + \frac{2}{3}|\sigma| - |y_\sigma|, \) since \( |\sigma| > \frac{2}{3}d(c_j) \). We also note that \( t \) cannot extend to the left beyond \( X'_\sigma \), or we could simply shift it over \( d(c_j) \) to the right. Therefore, the left end of \( t \) is in \( X'_\sigma \). The right end of \( t \) must also be within \( d(c_j) \) of the right end of \( s \), or between points A and B marked in Fig. 9. Similarly, the right

---

Fig. 9. Case 2 of Lemma 4.13. Determining the range of possible \( t \) and \( u \).
end of \( u \) is in \( X'_s \), and the left end may be anywhere within \( d(c_j) \) to the right of the left end of \( s \).

Because each string in \( c_j \) must have at least one \( \frac{2}{3} \)-repeater, let \( \sigma' \) be the smallest \( \frac{2}{3} \)-repeater of \( t \), and \( \sigma'' \) 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 \( X_\sigma \) or linked with it. The remaining cases which we consider all have \( \min\{E_r(s, \sigma), E_r(s, \sigma')\} > \frac{2}{3}d(c_j) + \frac{2}{3}|\sigma| - |y_\sigma| \) and are determined by whether \( t = u \) and whether \( X_{\sigma'} \) and \( X_{\sigma''} \) are linked with or nested within \( X_\sigma \).

In order to simplify our analysis, we will often assume that a string with a repeater \( \sigma \) extends from the left end of one copy of \( y_\sigma \) to the right end of another copy of \( y_\sigma \). 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 \( y_\sigma \) 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.

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

\begin{equation}
E_r(s, \sigma) + E_r(s, \sigma') = |y_\sigma| + 2d(c_j) - |y_\sigma|.
\end{equation}  \( \text{(8)} \)

Now consider extending \( t \) to the right. Any string \( t' \) which begins within \( d_{\text{pref}}(t, s) \) of the beginning of \( t \) must end before \( s \) due to the no-substring assumption, and we will only need to extend \( t \) by \( d_{\text{off}}(y_\sigma', y_\sigma) \), to the end of \( X_\sigma \) (see Fig. 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 \textit{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 \( E_r(t, \sigma') \leq d(c_j) - E_r(s, \sigma) \) and \( E_r(t, \sigma') \leq d(c_j) - E_r(s, \sigma) \):
be the smallest $\frac{2}{3}$-repeater of $v$. By our conditions on where $v$ starts and ends, $X_{\sigma_2}$ must be linked with $X_{\sigma}$ and contain $X_{\sigma'}$ as shown. We know by our choice of $s$ and $\sigma$ that $|\sigma| > |\sigma_2|$. By Lemma 4.6, $|\sigma_2| > \frac{1}{2}d(c_j)$, so we apply Lemma 4.8 to conclude that

$$|v_0| < \frac{3}{2}d(c_j).$$

(9)

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

$$\text{Cost}(t, \sigma') = |y_{\sigma'}| + d(c_j) - E_r(s, \sigma) + |\sigma'|$$

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

$$- |y_{\sigma} - E_r(s, \sigma)| + |\sigma'|$$

(Eq. (8))

$$= |y_{\sigma}| + E_r(s, \sigma) - d(c_j) + |\sigma'|$$

$$< |y_{\sigma}| + |\sigma'|$$

($E_r(s, \sigma) < d(c_j)$)

$$< \frac{3}{2}(d(c_j) + |\sigma'|)$$

(Eq. (9)).

Finally, suppose that there is an interloper in each direction, say $w$ and $v$ with $\frac{2}{3}$-repeaters $\sigma_1$ and $\sigma_2$, respectively, as in Fig. 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 \(X_{\sigma_1}\) and \(X_{\sigma_2}\) are linked, as are \(X_{\sigma_1}\) and \(X_{\sigma}\). We derive a lower bound on \(|\sigma|\):

\[
|\sigma| > |\sigma_1| + |y_\sigma| - d(c_j) \quad \text{(Lemma 4.8)}
\]
\[
> |\sigma_2| + |y_{\sigma_1}| - d(c_j) + |y_\sigma| - d(c_j) \quad \text{(Lemma 4.8)}
\]
\[
> |y_{\sigma_1}| + |y_\sigma| - \frac{3}{2} d(c_j) \quad \text{(Lemma 4.6)}
\]

\[
\Rightarrow \frac{1}{3} |\sigma| > \frac{2}{3} |\sigma_1| - \frac{1}{6} d(c_j) \quad \text{(Definition 3.2)}
\]
\[
\Rightarrow |\sigma_1| < \frac{1}{4} d(c_j) + \frac{1}{2} |\sigma|.
\]

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

\[
d_{\text{pref}}(y_{\sigma_1}, y_{\sigma}) < |y_{\sigma_1}| - d(c_j).
\] (10)

We use Eq. (10), Lemma 4.8, and Eq. (4.3) to bound this quantity:

\[
d_{\text{pref}}(y_{\sigma_1}, y_{\sigma}) < |y_{\sigma_1}| - d(c_j)
\]
\[
< |\sigma_1| - |\sigma_2|
\]
\[
< \frac{1}{3} d(c_j) + \frac{1}{2} |\sigma| - |\sigma_2|
\]
\[
< \frac{1}{2} |\sigma| - \frac{1}{4} d(c_j).
\]

We now calculate the anticipated cost of extending \(t\) to the left (in the direction of \(\sigma_1\), the larger of \(\sigma_1\) and \(\sigma_2\)):

\[
\text{Cost}(t, \sigma') \leq |y_{\sigma'}| + E_\ell(t, \sigma') + |\sigma'|
\]
\[
\leq |y_{\sigma'}| + d(c_j) - E_\ell(s, \sigma) + g_1 + |\sigma'|
\]
\[
< |y_{\sigma'}| + d(c_j) - \left(\frac{3}{2} d(c_j) + \frac{3}{2} |\sigma| - |y_\sigma|\right) + g_1 + |\sigma'| \quad \text{(Case bound)}
\]
\[
= |y_{\sigma'}| - \frac{11}{12} d(c_j) - \frac{1}{6} |\sigma| + |y_\sigma| + |\sigma'|
\]
\[
< |y_{\sigma'}| - \frac{11}{12} d(c_j) - \frac{1}{6} |\sigma| + (|\sigma| - |\sigma_1| + d(c_j)) + |\sigma'| \quad \text{(Lemma 4.8)}
\]
\[
= |y_{\sigma'}| + \frac{1}{12} d(c_j) + \frac{5}{6} |\sigma| - |\sigma_1| + |\sigma'|.
\]
In the last inequality above we were able to apply Lemma 4.8 because $X_{\sigma}$ and $X_{\sigma_1}$ are linked; now we can apply it again, because $X_{\sigma_1}$ and $X_{\sigma_2}$ are also linked.

\[
\begin{align*}
\text{Definition 3.2} & \quad \Rightarrow \\
<yd\sigma'>' + hd(cj) & \quad + \frac{5}{6}|\sigma| - \frac{5}{6}|\sigma_2| + \frac{5}{6}|\sigma_1| + |\sigma'| \\
& \quad \leq gd(cj) + \frac{1}{2}|\sigma'| + \frac{1}{2}|\sigma_1| - |y_\sigma'|
\end{align*}
\]

This concludes the analysis of Case 2A.

In the remaining two cases, $t \neq u$; that is, $LString(c, \sigma) \neq RString(c, \sigma)$. Let $\sigma'$ be the smallest $\frac{\epsilon}{2}$-repeater of $t$ and $\sigma''$ be the smallest $\frac{\epsilon}{3}$-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 $X_{\sigma}$, we observe that $E_r(s, \sigma) = d_{pref}(y', y)$ and $E_r(s, \sigma) = d_{suff}(y'', y)$, so we can apply Lemma 4.11 and conclude that $\min\{E_r(s, \sigma), E_r(s, \sigma)\} < \frac{1}{3}d(cj) + \frac{1}{3}|\sigma| - |y_\sigma|$. 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 $X_{\sigma}$ (Case 2B), and when only $X_{\sigma''}$ is linked with $X_{\sigma}$ (Case 2C).

Case 2B: $\min\{E_r(s, \sigma), E_r(s, \sigma)\} > \frac{1}{3}d(cj) + \frac{1}{3}|\sigma| - |y_\sigma|$, $t \neq u$, $X_{\sigma'}$ and $X_{\sigma''}$ both nested. We show that $t$ can be extended to the right within our bounds (see Fig. 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 \( X_\sigma \). We use Lemma 4.10, the Case bound on \( E_r(s, \sigma) \) and \( E_r(s, \sigma) \), and the fact that \( E_r(s, \sigma) + E_r(s, \sigma) = |y_\sigma| + 2d(c_j) - |y_\sigma| \):

\[
\text{Cost}(t, \sigma') \leq |y_\sigma| + E_r(t, \sigma') + |\sigma'|
\]

\[
\leq |y_\sigma| + d_{\text{suffix}}(y', y) + |\sigma'|
\]

\[
= |y_\sigma| + (|y_\sigma| - |y_\sigma| - d_{\text{pref}}(y', y)) + |\sigma'|
\]

\[
= |y_\sigma| + E_r(s, \sigma) - d(c_j) + |\sigma'|
\]

\[
= \frac{5}{3}|\sigma'| + |y_\sigma| + E_r(s, \sigma) - d(c_j) - \frac{2}{3}|\sigma'|.
\]

We apply Lemma 4.10 and the fact that \( E_r(s, \sigma) = d_{\text{pref}}(y', y) \) to bound the last term above,

\[
\leq \frac{5}{3}|\sigma'| + |y_\sigma| + E_r(s, \sigma) - d(c_j) - \frac{1}{3}(E_r(s, \sigma) + |y_\sigma| - 2d(c_j))
\]

\[
= \frac{5}{3}|\sigma'| + \frac{1}{3}|y_\sigma| + \frac{1}{3}E_r(s, \sigma) + \frac{1}{3}d(c_j)
\]

\[
< \frac{5}{3}|\sigma'| + \frac{4}{3}d(c_j) \quad \text{(Eq. (5))}
\]

\[
< \frac{5}{3}(d(c_j) + |\sigma'|).
\]

Because \(|\sigma'| > |\sigma''|\) and \( \sigma' \) 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 Fig. 11b. Let \( \sigma_2 \) 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

\[
|\sigma_2| < |\sigma| - |y_\sigma| + d(c_j). \quad (11)
\]

We bound the distance that the interloper \( w \) extends beyond \( X_\sigma \):

\[
d_{\text{suffix}}(y, \sigma_2) < |y_{\sigma_2}| - d(c_j) - (E_r(s, \sigma) - 2d(c_j) - |y_\sigma|)
\]

\[
= |y_{\sigma_2}| + d(c_j) - E_r(s, \sigma) - |y_\sigma|.
\]

We now calculate the cost of extending \( t \) to the right:

\[
\text{Cost}(t, \sigma') \leq |y_\sigma| + E_r(t, \sigma') + |\sigma'|
\]

\[
\leq |y_\sigma| + d_{\text{suffix}}(y_{\sigma'}, y_{\sigma_2}) + d_{\text{suffix}}(y, y_{\sigma_2}) + |\sigma'|
\]

\[
= |y_\sigma| + (|y_\sigma| - (d(c_j) - E_r(s, \sigma)) - |y_{\sigma'}|)
\]

\[
+ d_{\text{suffix}}(y, y_{\sigma_2}) + |\sigma'|
\]
Case 2C: \[ \min\{E_r(s, \sigma), E_r(s, \sigma')\} > \frac{3}{2}d(c_j) + \frac{3}{2}|\sigma| - |y_{\sigma}|, \quad t \neq u, \quad X_{\sigma''} \text{ (but not } X_{\sigma'}) \text{ linked with } X_{\sigma}. \]

In this case \( X_{\sigma''} \) might be nested within \( X_{\sigma''} \) (Fig. 12a), or not (Fig. 12b). It is an unlikely case to give us trouble, because here the smaller \( \frac{3}{2} \)-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 \[ |\sigma''| > \frac{1}{2}d(c_j). \]

Since Lemma 4.8 also applies we have

\[ |y_{\sigma}| < |\sigma| - |\sigma''| + d(c_j) < \frac{3}{2}d(c_j). \]  

(12)

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

\[
\begin{align*}
\text{Cost}(t, \sigma') & \leq |y_{\sigma'}| + E_r(t, \sigma') + |\sigma'| \\
& \leq |y_{\sigma'}| + d_{\text{suff}}(y_{\sigma'}, y_{\sigma}) + |\sigma'| \\
& \quad - |y_{\sigma'}| + (|y_{\sigma'}| - |y_{\sigma'}| - (d(c_j) - E_r(s, \sigma))) + |\sigma'| \\
& < |y_{\sigma}| + |\sigma'| \quad \text{(Eq. (12))}
\end{align*}
\]

(12)
Now suppose there was an interloper $v$ with smallest $\frac{2}{3}$-repeater $\sigma_2$. Then $X_{\sigma_2}$ would be linked with $X_{\sigma''}$ and $X_{\sigma}$, and Lemma 4.11 would apply as in Fig. 7(d), and we would once again be in Case 1.

Subcase (ii): Now $X_{\sigma''}$ is not nested within $X_{\sigma'}$, as in Fig. 12b. If there are no interlopers, then we only have to extend $t$ to the right to the end of $X_{\sigma}$:

$$\text{Cost}(t, \sigma') \leq |y_{\sigma'}| + E_r(t, \sigma') + |\sigma'|$$

$$\leq |y_{\sigma'}| + E_r(s, \sigma) + d(c_j) - |y_{\sigma'}| - (2d(c_j) - |y_{\sigma}|) + |\sigma'|$$

$$= |y_{\sigma}| + E_r(s, \sigma) - 2d(c_j) + |\sigma'|.$$

We apply Lemma 4.10 to complete the analysis:

$$\text{Cost}(t, \sigma') < |y_{\sigma}| + (|\sigma'| - |y_{\sigma}| + 2d(c_j)) - d(c_j) + |\sigma'|$$

$$= 2|\sigma'| - d(c_j)$$

$$< \frac{2}{3}(d(c_j)) + |\sigma'|.$$

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

This completes the proof of Case 4, which completes the proof of the lemma. $\square$

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 $1 + \varepsilon$ for some very small constant $\varepsilon$. There is also a gap between the best proven bound on the performance of a simple greedy algorithm for the problem [4], 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
