Short Superstrings and the Structure of Overlapping Strings

Chris Armen* Clifford Stein †

Department of Computer Science
Dartmouth College
Hanover, NH

June 14, 1995

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 computational biology and data compression. The shortest superstring problem is NP-hard [6]; in fact, it was recently shown to be MAX SNP-hard [1]. Given the importance of the applications, several heuristics and approximation algorithms have been proposed. Constant factor approximation algorithms have been given in [1] (factor of 3), [16] (factor of $2\frac{8}{9}$), [2] (factor of $2\frac{5}{6}$) and [10] (factor of $2\frac{50}{63}$).

Informally, the key to any algorithm for the shortest superstring problem is to identify sets of strings with large amounts of similarity, or *overlap.* While the previous algorithms and their analyses have grown increasingly sophisticated, they reveal remarkably little about the structure of strings with large amounts of overlap. In this sense, they are solving a more general problem than the one at hand.

In this paper, we study the structure of strings with large amounts of overlap and use our understanding to give an algorithm that finds a superstring whose length is no more than $2\frac{3}{4}$ times that of the optimal superstring. Our algorithm runs in $O(|S| + n^3)$, which matches that of previous algorithms. We prove several interesting properties about short periodic 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.

***Contact author.** Email address: `armen@cs.dartmouth.edu`.

†Email address: `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.

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 [3, 11, 13] and data compression [6, 14]. 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 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 other major feature of the application which is not captured by the shortest superstring problem is that the laboratory data may contain errors of up to 5%, while in the shortest superstring problem we are only concerned with exact matches.

The shortest superstring problem is NP-hard [6]; in fact, it was recently shown to be MAX SNP-hard [1]. Given the importance of the applications, several heuristics and approximation algorithms have been proposed. One often used algorithm is a greedy algorithm that repeatedly merges the pair of strings with the maximum amount of overlap. Turner [17] and Tarhio and Ukkonen [15] independently proved that the greedy algorithm constructs a superstring that achieves at least half as much overlap as an optimal superstring. However, this does not guarantee a constant approximation with respect to the length of the resulting superstring. When the amount of overlap in the optimal solution is very large, a superstring which achieves half of the optimal overlap may be longer than the optimal solution by a factor of $\Theta(n)$.

The first provable bound on the length approximation of the greedy algorithm was provided by Blum et al.[1], who showed that the greedy algorithm returns a string that is no longer than four times optimal; they also give a modified greedy algorithm that returns a string that is within three times optimal. Teng and Yao [16] gave a nongreedy algorithm that finds a string that is within $2\frac{8}{9}$ of optimal. Independently of our work, Czumaj et al.[2] refined the Teng algorithm to achieve a $2\frac{5}{6}$ approximation. Recently a new result for the maximum traveling salesman problem has been used to improve the algorithm of [1] to obtain an approximation slightly better than 2.8 [10].

The algorithms in [16, 2, 10] share the graph representation and framework of the 3-approximation of Blum et al. as a point of departure, as does our own. However, these other improvements are largely graph-theoretic; in contrast, our approach captures a great deal of the structure of the problem which is not evident in the graph representation. Informally, the key to any algorithm for the shortest superstring problem is to identify sets of strings with large amounts of similarity, or *overlap*. We prove several key properties of such strings, and exploit these properties to construct a superstring whose length is no more than $2\frac{3}{4}$ times that of the optimal superstring.

We believe that our approach is important beyond the fact that we have achieved the best approximation ratio to date. Because we are dealing more specifically with the properties of strings, we feel our approach will be more readily adaptable to variations of the problem that arise in applications. We also believe that the machinery which we introduce to study the structure of overlapping strings will yield a significantly better ratio for this problem, and may have application to other string problems.

We now give a brief overview of our approach. All 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 $\mathrm{ov}(u,v)$ 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 [1] uses a theorem about infinite periodic functions [4], 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 interesting 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. We give more details in later sections.

Our algorithm runs in $O(|S|+n^3)$ time, as do the other $O(1)$-approximation algorithms when properly implemented.

2 Preliminaries

For consistency, we use some notation and definitions of [1] and [16]. 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 $t = xv$. We call u the *prefix* of s with respect to t, $\mathrm{pref}(s,t)$, and refer to $|u|$ as the distance from s to t, $\mathrm{d}(s,t)$. Observe that for any s and t, $\mathrm{ov}(s,t) + \mathrm{d}(s,t) = |s|$. 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 $\mathrm{d}(s_i,s_j)$ 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 $\mathrm{pref}(s_i,s_j) \leq |s|$, we can conclude that $\mathrm{opt}(TSP) \leq \mathrm{opt}(S)$, where $\mathrm{opt}(TSP)$ is the optimal solution to TSP defined above. This TSP is directed (sometimes called *asymmetric*), so the best known approximation [5] is only within a factor of $O(\log n)$. Thus, we must exploit more

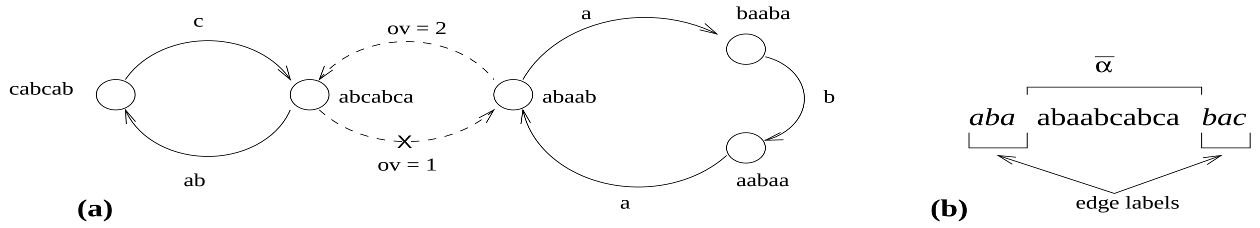

Figure 1: Execution of GENERIC SUPERSTRING ALGORITHM. Nodes are labeled with strings, edges with prefixes. (a) The graph after Step (4). Solid edges are in C, dashed edges in CC. The edge with an X is the one discarded in Step (4). (b) The final string consisting of $\bar{\alpha}$ (the merge of *abaab* and *abcabca*) along with the labels from the edges of the cycles.

of the structure of the problem in order to achieve a better approximation for length.

Given a directed graph G, with weights on the edges, a *cycle cover* is a set of cycles such that each vertex is in exactly one cycle. A minimum-cost cycle cover C is a cycle cover such that the sum of the weights of the edges in all the cycles is minimized. A minimum-cost cycle cover can be computed in $O(n^3)$ time by a well-known reduction to the assignment problem [12]. Let $w(C)$ be the sum of the weights of the edges of C. Since a tour is a cycle cover, $w(C) \leq \text{opt}(TSP)$. When we say that a string s_i is in some cycle c of cycle cover 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.

Because $\text{ov}(s_i, s_j) + \text{d}(s_i, s_j) = |s_i|$, one could also weight the edges by their overlap, find a maximum-cost cycle cover and obtain the same solution. A superstring which 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. [17, 16, 10]) than approximating the shortest superstring.

We now describe a generic superstring algorithm used, in some form, by [1],[16] and [2]. An execution of the algorithm appears as Figure 1.

<u>GENERIC SUPERSTRING ALGORITHM</u>

1) Find a minimum cost cycle cover C in the distance graph.
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) Break each cycle $\gamma \in CC$ by deleting one edge.
5) Concatenate the remaining strings arbitrarily.
6) Extend each representative r_c by the concatenation of the prefixes around c.

The first cycle cover identifies sets of strings that have large amounts of overlap. This allows us to form the second cycle cover, in which approximating overlap and the string length are roughly comparable, so stronger bounds apply. Step (6) correctly extends the superstring for R into a superstring for S, as proved in [16].

We now analyze the generic algorithm in a way that anticipates our improvements. A more detailed analysis appears in [1]. Let $\text{d}(C')$ be the sum of the distances and $\text{ov}(C')$ be the sum of the overlaps of the edges in a cycle cover C'. Consider the second cycle cover CC. Let $\mathrm{opt}(R)$ be the optimal superstring on the strings in $r_c \in R$ and observe that $\mathrm{opt}(R) \leq \mathrm{opt}(S)$. Let $\bar{\alpha}$ be the string produced in Step 5, a superstring of R, and let $\mathrm{opt}(\mathrm{ov}(R)) = |R| - \mathrm{opt}(R)$ be the amount of overlap in the optimal superstring for R. Since the shortest superstring for R is a cycle cover for G', $\mathrm{ov}(CC) \geq \mathrm{opt}(\mathrm{ov}(R))$. However, the superstring $\bar{\alpha}$ does not have as much overlap as CC, since we delete one edge from each cycle.

For a cycle γ, let ov^n_γ denote the overlap in the edge deleted in Step 4, and let ov^t_γ denote the remaining overlap in γ. Let $\mathrm{ov}^t = \sum_{\gamma \in CC} \mathrm{ov}^t_\gamma$ and $\mathrm{ov}^n = \sum_{\gamma \in CC} \mathrm{ov}^n_\gamma$. Thus, $|\bar{\alpha}| \leq |R| - \mathrm{ov}^t$. By definition, $|R| \leq \mathrm{opt}(R) + \mathrm{opt}(\mathrm{ov}(R)) \leq \mathrm{opt}(R) + \mathrm{ov}(CC)$. Combining these two inequalities with $\mathrm{ov}(CC) = \mathrm{ov}^n + \mathrm{ov}^t$, gives that $\bar{\alpha} \leq \mathrm{opt}(R) + \mathrm{ov}^n$. We then must extend each cycle, in Step 6. Let $\mathrm{Ext}(\gamma)$ be the cost of extending all cycles $c \in C$ s.t. $r_c \in \gamma$. Then we can express the length of α, the string obtained, as

$$|\alpha| \leq \mathrm{opt}(R) + \sum_{\gamma \in CC} \left(\mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) \right). \tag{1}$$

Let $\mathrm{d}(c)$ be the sum of the weights of the edges of a cycle $c \in C$; so $\mathrm{d}(C) = \sum_{c \in C} \mathrm{d}(c)$. To obtain a 3-approximation, observe that the set of edges which contribute to ov^n form a matching M on G'. Now we employ a key lemma from [1]:

Lemma 2.1 ([1]) *Let c, c' be cycles in a minimum cycle cover C with strings $s \in c$ and $s' \in c'$. Then the overlap between s, s' is less than $d(c) + d(c')$.*

Since M is a matching, each cycle c is at an endpoint of a string at most once, and hence $\mathrm{ov}^n \leq \mathrm{d}(C)$. Now, we extend $\bar{\alpha}$ by the edge labels on each cycle, adding a total of $\mathrm{d}(C)$ to the length of the string. Let α be the resulting string. We conclude that

$$|\alpha| \leq \mathrm{opt}(R) + \sum_{\gamma \in CC} \mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) \leq \mathrm{opt}(R) + \mathrm{d}(C) + \mathrm{d}(C) \leq 3\mathrm{opt}(S)$$

since both $\mathrm{d}(C)$ and $\mathrm{opt}(R)$ are lower bounds on $\mathrm{opt}(S)$.

The analysis above makes it clear that the cycle cover CC actually partitions the cycles in the cycle cover C, and hence each cycle in CC can be analyzed separately. As was observed by [16] in their $2\frac{8}{9}$ algorithm, if γ has three or more vertices, then $\mathrm{ov}^n_\gamma \leq \frac{2}{3} \sum_{c \in \gamma} \mathrm{d}(c)$.

Thus we can restrict our attention to 2-cycles in CC, if we are to get an approximation bound of than $2\frac{3}{4}$. We will analyze each 2-cycle in CC separately, and obtain a $2\frac{3}{4}$ bound by proving structural properties of these 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 algorithm above, that, for each 2-cycle γ in CC consisting of vertices v and t, attains a bound of $ov^n_\gamma + Ext(\gamma) \leq \beta(d(c_v) + d(c_t))$, for some $\beta \geq \frac{5}{3}$, is a $(1+\beta)$-approximation algorithm for the shortest superstring problem.*

We will show in Section 4 that for each 2-cycle $\gamma \in CC$, either ov^n_γ or $\mathrm{Ext}(\gamma)$ can be bound better than in the above analysis.

In order to talk about the structure of cycles, we define a few terms. The reader is referred to [1] for a more complete discussion. We call a string s *irreducible* if all cyclic shifts of s

yield unique strings, and *reducible* otherwise. Let gen(ς) be the string formed by an infinite repetition of a string ς. We say that s has *periodicity* x if there exists a string ς with $|\varsigma| = x$ such that s is substring of gen(ς). Let per(c) be the string formed by concatenating all the labels of the edges on a cycle c, then for each string $s \in c$, s is a substring of gen(per(c)). We define $\overset{\infty}{\mathrm{per}}(c)$ to be gen(per(c)). Note that per(c) must be irreducible; otherwise a cycle with less total distance could generate the same strings, contradicting the minimality of the cycle cover. The irreducibility of the periods of cycles in a minimum cycle cover will figure prominently in many of our proofs.

We can now state a corollary to Lemma 2.1 that we will also use frequently in our proofs.

Corollary 2.3 ([4]) *Let w be a substring of both gen(σ_j) and gen(σ_k). Then if $|w| \geq |\sigma_j| + |\sigma_k|$, either σ_j or σ_k is reducible.*

3 The Structure of High-Overlap 2-Cycles

In the previous section we saw that in order to obtain a better approximation for the shortest superstring problem it is sufficient to perform better in the presence of 2-cycles in the second cycle cover of the generic superstring algorithm. In this section we present structural lemmas concerning 2-cycles which are the key to our approach.

Suppose we choose v and t as representatives of two cycles of the first cycle cover C, and they form a 2-cycle in CC in which one of ov(v, t) or ov(t, v) is large but the other is small. In Step #4 we will break the 2-cycle to form a string, and since 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 discard one of them. In 3-or-larger cycles this does not cost us more than one-third of the overlap in the cycle; in a 2-cycle we will lose up to half of the overlap, which is the worst case of the generic algorithm. We observe that both edges cannot participate in an optimal solution; in this sense the second cycle cover has achieved "false overlap".

Our strategy is to anticipate, when we choose representatives, the potential of each string to participate in a high-overlap 2-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 representative. Otherwise we know a great deal about the structure of the entire cycle and can trade off 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 must have both a significant prefix and suffix with some smaller period, which might correspond to the period of another cycle in the cover and hence some other representative. We require some notation to describe this potential.

Definition 3.1 Let z be a string in cycle c and let σ be an irreducible string with $|\sigma| < \mathrm{d}(c)$. Then σ is a *(g, h)-repeater of z* if there exist witnesses y_ℓ and y_r, such that

- y_ℓ is a prefix of z and y_r is a suffix of z
- y_ℓ and y_r are substrings of gen(σ)

- $|y_\ell|, |y_r| > g\mathrm{d}(c) + h|\sigma|$.

Consider the string z in Figure 2b and let $g = h = \frac{3}{4}$. Here $\mathrm{per}(c) = ababadababadab$, $\sigma = ababad$, $y_\ell = ababadababadababa$ and $y_r = ababadababadabab$. So $|y_\ell|, |y_r| > \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma|$, and we say that σ is a $(\frac{3}{4}, \frac{3}{4})$-repeater of z.

Note that in our example y_ℓ and y_r are almost the same; this is not a complete coincidence. All the repeaters we will be considering in this paper will have $g \geq \frac{1}{2}$ and hence y_ℓ and y_r *must* overlap, often significantly (as in this example). For convenience we will define one witness y_σ which contains both y_ℓ and y_r.

Definition 3.2 Let σ be a (g, h)-repeater in a cycle c with $g \geq \frac{1}{2}$. The maximal witness y_σ is the maximum-length substring of $\mathrm{gen}(\sigma)$ that is also a substring of $\overset{\infty}{\mathrm{per}}(c)$.

In other words, if we take σ and try to repeat it as many times as possible, in both directions, while being consistent with c, we get y_σ. In the example above $y = y_\ell$. When the context is clear, we will drop the σ and just refer to witness y.

Henceforth when discussing and proving properties of cycles, we will refer to the maximal witness y_σ rather than to the underlying pair of witnesses y_ℓ and y_r. This simplification is conservative: in our analysis we either use the lower bound on y_σ from the definition of (g, h)-repeaters or the upper bound just proved. Both of these apply to the actual witness pair y_ℓ and y_r as well as to the maximal witness y_σ.

We obtain a simple upper bound on the length of y:

Lemma 3.3 *Let y be a maximal witness for some (g, h)-repeater σ in a cycle c. Then $|y| < d(c) + |\sigma| < 2d(c)$.*

Proof: Observe that by definition both σ and per(c) are irreducible, and $|\sigma| < \mathrm{d}(c)$. Suppose $|y| > \mathrm{d}(c) + |\sigma|$ and apply Corollary 2.3 to derive a contradiction. ■

The idea behind (g, h)-repeaters is to identify periodic substrings of the period of a cycle in C. We will also be interested in identifying that portion of a cycle that is *not* consistent with some (g, h)-repeater σ. Note that a copy of y_σ begins every $\mathrm{d}(c)$ in $\overset{\infty}{\mathrm{per}}(c)$.

Definition 3.4 Let c be a cycle with (g, h)-repeater σ and maximal witness y. Fix a copy of y in $\overset{\infty}{\mathrm{per}}(c)$. The point just to the left of the first character of y is the *leftish 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 *rightish* of σ. The *characteristic* of σ, X_σ, is the interval from the leftish to the rightish. 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. Figure 2b 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 leftish and rightish of adjacent copies of X_σ. A negative characteristic appears in Figure 2a and can be pictured as a single solid entity (perhaps of size zero) which 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 $\mathrm{per}(c)$ which must be included as a prefix and suffix of some string z if z is to participate in a high-overlap 2-cycle.

$z =$	$\underline{\text{abababrst}}$ababab
	) (
$y = y_\ell =$	ababab
$y_r =$	ababab
$\sigma =$	ab

(a)

$z =$	$\underline{\text{ababadababadab}}$ababadababadabab
	( ) ( ) (
$y = y_\ell =$	ababadababadababa
$y_r =$	ababadababadabab
$\sigma =$	ababad

(b)

Figure 2: Positive and Negative Characteristics. Per(c) is underlined. (a) shows a negative characteristic. (b) shows a positive characteristic. y and σ are also shown.

Recall that we defined (g, h)-repeaters (Definition 3.1) in terms of some string z in a cycle c which contained witnesses y_ℓ and y_r as a prefix and suffix. In general there might be several such strings in c which could satisfy the definition. We say that σ is *active* in each of these strings.

Lemma 3.5 *Let t be a string in cycle c and v be a string in cycle c'. If $ov(v,t) \geq gd(c) + hd(c')$ and $ov(t,v) \geq gd(c) + hd(c')$, then there is a (g,h)-repeater $\sigma = per(c')$ which is active in t.*

Proof: Observe that per(c') satisfies each requirement of Definition 3.1 as follows. Both per(c) and per(c') are irreducible because they are periods of cycles in a minimum cover. The witnesses y_ℓ and y_r are exactly the prefix and suffix of t which overlap with v. The witnesses are sufficiently long by the overlap bound as stated in the lemma. ■

We can identify the possible relationships which the various characteristics may have with respect to each other. We say that two characteristics X_{σ_i}, X_{σ_j} are *properly nested* if X_{σ_i} is a positive characteristic and X_{σ_j} falls strictly within X_{σ_i}. We say that two characteristics X_{σ_i}, X_{σ_j} are *disjoint* if their intervals are disjoint.

In order to obtain an approximation ratio of $2\frac{3}{4}$ we will be particularly interested in $(\frac{3}{4}, \frac{3}{4})$-repeaters. We will show that any two positive characteristics of $(\frac{3}{4}, \frac{3}{4})$-repeaters are either nested or disjoint. This implies that any set of positive characteristics is well-parenthesized. We will also show that there can be no more than two $(\frac{3}{4}, \frac{3}{4})$-repeaters with negative characteristics in a cycle.

We will call X_σ a *major characteristic* if it is not nested within any other characteristic. A *minor characteristic* X_σ is one which is nested within some positive characteristic. We will say that σ is a *major (minor) repeater* if X_σ is a major (minor) characteristic.

This parenthesis structure naturally gives rise to a forest representation of the nesting of minor characteristics within major ones. We can view each major characteristic as the root of a parse tree, and each minor characteristic as another node in the tree according to its nested position, as pictured in Figure 3. Parent-child relationships are thus well-defined and we have a tree for each major characteristic. We will use this representation in our algorithm to handle minor characteristics.

For any string in any cycle c, we can prove properties about the structure and size of possible (g, h)-repeaters and their characteristics. The following four lemmas provide this characterization, which shows that there can be only a few small repeaters and that all repeaters are nicely parenthesized. We will use these lemmas in the analysis of our new algorithms.

Lemma 3.6 *Let c be an irreducible cycle. Then there is at most one σ such that σ is a $(\frac{1}{2}, 2)$-repeater of c.*

Proof: Suppose for purpose of contradiction that there are two distinct $(\frac{1}{2}, 2)$-repeaters, σ and σ'. Consider their maximal witnesses y and y'. By the definition of a $(\frac{1}{2}, 2)$-repeater, we know

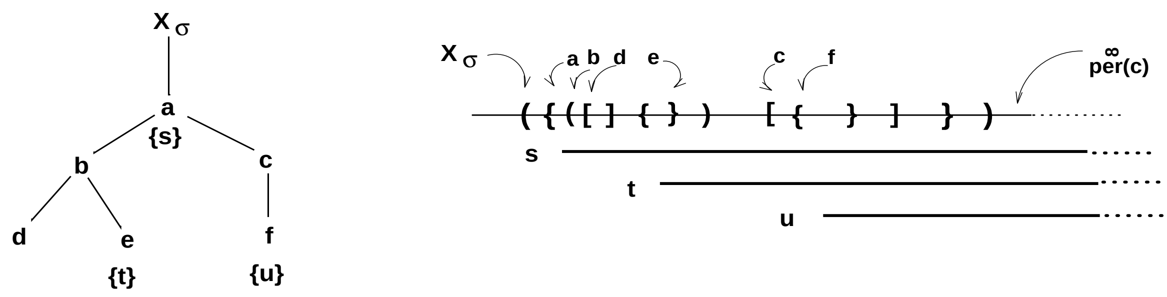

Figure 3: Parse Tree Representation of Nested Characteristics with Associated Strings.

$$|y| \geq \frac{1}{2}\mathrm{d}(c) + 2|\sigma| \quad \text{and} \quad |y'| \geq \frac{1}{2}\mathrm{d}(c) + 2|\sigma'|.$$

These lower bounds on length guarantee that

$$\mathrm{ov}(y, y') + \mathrm{ov}(y', y) \geq |y| + |y'| - \mathrm{d}(c) \geq 2|\sigma| + 2|\sigma'|.$$

It follows that $\max(\mathrm{ov}(y, y'), \mathrm{ov}(y', y)) \geq |\sigma| + |\sigma'|$. We complete the proof by applying Corollary 2.3. ■

Lemma 3.7 *Let c be an irreducible cycle. Then there are at most two $(\frac{3}{4}, \frac{3}{4})$-repeaters of c, σ_1 and σ_2, such that $|\sigma_i| \leq \frac{1}{3}d(c), i \in \{1, 2\}$.*

Proof: Suppose for purpose of contradiction that there are three such repeaters σ_1,σ_2, and σ_3 with maximal witnesses y_1,y_2, and y_3. We will sum the overlaps of the witnesses, and show that at least one overlap will be larger than the respective $|\sigma_i| + |\sigma_j|$, leading to a contradiction by Corollary 2.3.

Consider the three overlaps in the order $\mathrm{ov}_1 = \mathrm{ov}(y_1, y_2)$, $\mathrm{ov}_2 = \mathrm{ov}(y_2, y_3)$, $ov_3 = \mathrm{ov}(y_3, y_1)$. We know that the overlap between two witnesses y_i and y_j is $\mathrm{ov}_i = |y_i| - \mathrm{d}(y_i, y_j)$. Summing the three overlaps and applying the definition of $(\frac{3}{4}, \frac{3}{4})$-repeater gives us

$$\begin{aligned}
\sum_{1 \leq i \leq 3} \mathrm{ov}_i &= \sum_{1 \leq i \leq 3} |y_i| - \mathrm{d}(c) \\
&\geq \frac{9}{4}\mathrm{d}(c) + \frac{3}{4} \sum_{1 \leq i \leq 3} |\sigma_i| - \mathrm{d}(c) \\
&= \frac{5}{4}\mathrm{d}(c) + \frac{3}{4} \sum_{1 \leq i \leq 3} |\sigma_i|
\end{aligned}$$

To avoid a contradiction by Corollary 2.3, we would need for each overlap between witnesses y_i and y_j that $\sigma_i + \sigma_j > \mathrm{ov}(y_i, y_j)$. Summing over the three overlaps we have

$$2 \sum_{1 \leq i \leq 3} |\sigma_i| > \frac{5}{4}\mathrm{d}(c) + \frac{3}{4} \sum_{1 \leq i \leq 3} |\sigma_i|$$

Solving for $\sum_{1 \leq i \leq 3} |\sigma_i|$ gives us $\sum_{1 \leq i \leq 3} |\sigma_i| > \mathrm{d}(c)$ which contradicts the condition of the Lemma that each $|\sigma_i| \leq \frac{1}{3}\mathrm{d}(c)$. ■

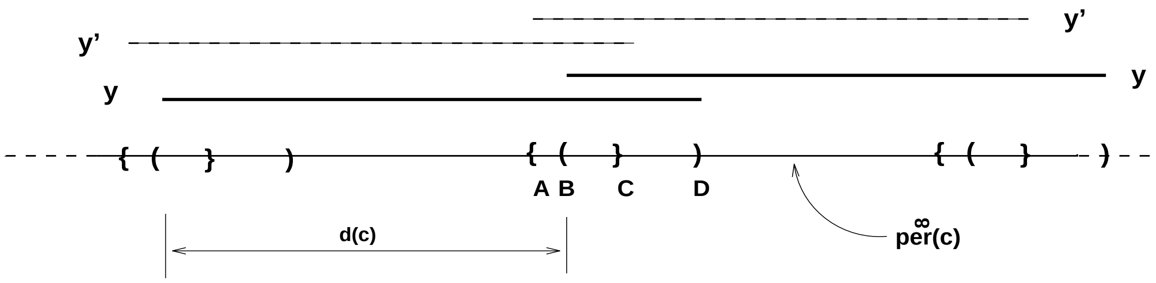

Figure 4: Proof of Lemma 3.9, Regions I,II,III.

When $|\sigma|$ is large relative to $\mathrm{d}(c)$ and $|y|$, we have no useful bound on the number of major or minor $(\frac{3}{4},\frac{3}{4})$-repeaters. However, as discussed above we can show that major and minor characteristics in this case are well parenthesized and thus are naturally represented as a forest. First we consider the case of a pair of negative characteristics.

Lemma 3.8 *Let σ,σ' be $(\frac{3}{4},\frac{3}{4})$-repeaters in cycle c, $|\sigma| > |\sigma'|$, with witnesses y, y'. If both $y, y' \leq$ $d(c)$, then X_σ and $X_{\sigma'}$ are disjoint.*

Proof: Assume for purpose of contradiction that there is such a pair which is not disjoint. If we fix the position of one of the characteristics, say without loss of generality X_σ, and require $X_{\sigma'}$ to overlap X_σ, then y and y' must be overlapped in such a way that they are completely contained within a distance of $\mathrm{d}(c)$. In order to avoid the contradiction resulting from the application of Corollary 2.3 it is necessary that

$$|\sigma| + |\sigma'| > \mathrm{ov}(y, y') \geq y + y' - \mathrm{d}(c) \geq \frac{1}{2}\mathrm{d}(c) + \frac{3}{4}(\sigma + \sigma').$$

Solving for $\sigma + \sigma'$ gives us $\sigma + \sigma' > 2\mathrm{d}(c)$. This contradicts the definition of $(\frac{3}{4},\frac{3}{4})$-repeater, which requires that $|\sigma| < \mathrm{d}(c)$. ■

Lemma 3.9 *Let σ,σ' be $(\frac{3}{4},\frac{3}{4})$-repeaters in cycle c, $|\sigma| > |\sigma'|$, with witnesses y, y'. If both $y, y' >$ $d(c)$, then X_σ and $X_{\sigma'}$ are disjoint or properly nested.*

Proof: Suppose for purpose of contradiction that the two positive characteristics X_σ and $X_{\sigma'}$ are neither disjoint nor properly nested, and further suppose without loss of generality that $|\sigma| > |\sigma'|$. We will consider six cases, conditioned on the relative values of $|\sigma|$ and $|\sigma'|$ in units of $\mathrm{d}(c)$. In each case we will generate a contradiction to our initial assumption that X_σ and $X_{\sigma'}$ are neither disjoint nor properly nested. The union of the six cases includes all possible values of $|\sigma|$ and $|\sigma'|$, as illustrated in Figure 5.

Region I: $|\sigma| \leq \mathrm{d}(c) - |\sigma'|$.

Figure 4 reflects the situation; in particular note that a copy of y and of y' must overlap by more than $\mathrm{d}(c)$ in $\overset{\infty}{\mathrm{per}}(c)$. We can apply Corollary 2.3 to this overlap, which gives us $|\sigma| + |\sigma'| > \mathrm{d}(c)$, contradicting our bound for Region I.

For the next four regions we introduce a technique which is of general utility in considering strings with multiple periods. This shift technique will derive a contradiction based on the following observation, which can easily be verified by the definition of maximal witness.

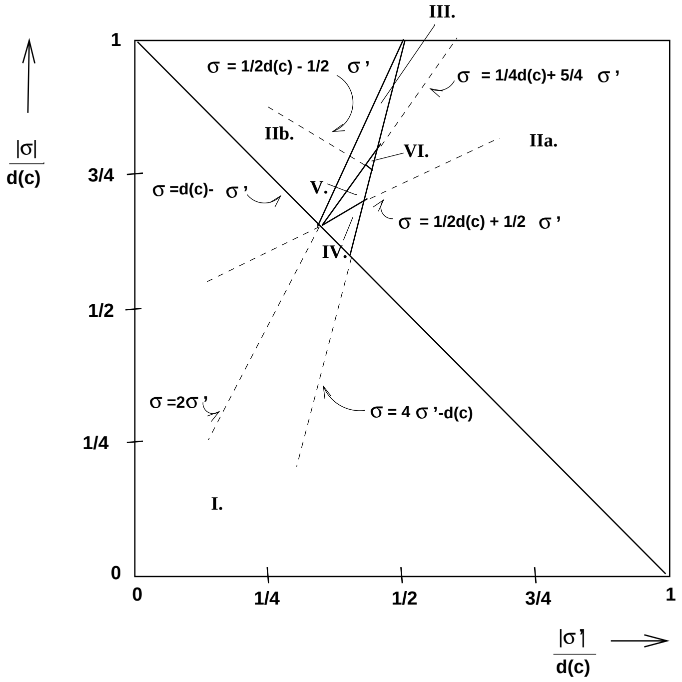

Figure 5: Regions of $|\sigma|$vs.$|\sigma'|$ in proof of Lemma 3.9. Solid lines denote region boundaries.

Observation 3.10 *Let y be the maximal witness for a (g,h)-repeater σ in a cycle c, and fix a copy y^* of y in $\overset{\infty}{\text{per}}(c)$. Index the character positions of $\overset{\infty}{\text{per}}(c)$ 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 3.10 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 y_σ, any two characters that are σ apart are identical, and in $\overset{\infty}{\text{per}}(c)$, 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 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.

Region IIa: $|\sigma| \leq 4|\sigma'| - \mathrm{d}(c)$.

Start: B			**Goal:** $+|\sigma|$
No.	*Move*	*Requirement*	*Comments*
1.	$-|\sigma'|$	$|\sigma'| < \mathrm{d}(c)$	Def. Repeater
2.	$+|\sigma|$	$|\sigma| - |\sigma'| < |y| - \mathrm{d}(c)$	See below
3	$+|\sigma'|$	$|\sigma| < \mathrm{d}(c)$	Def. Repeater

For move #2 to be valid, it is necessary that $|\sigma| - |\sigma'| < |y| - \mathrm{d}(c)$. Supposing the contrary, we get $|\sigma| - |\sigma'| > |y| - \mathrm{d}(c) > \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma| - \mathrm{d}(c)$. Solving for $|\sigma|$ we get $|\sigma| > 4|\sigma'| - \mathrm{d}(c)$, contradicting the range of Region IIa and proving that the shift is valid.

In this set of shifts we had only one, #2, which depended on particular values of $|\sigma|$,$|\sigma'|$,$|y|$ or $|y'|$ for its validity. We used a lower bound on $|y|$ to derive a contradiction for this region, but we also state the condition for later use:

$$|\sigma| > |\sigma'| + |y| - \mathrm{d}(c) \tag{2}$$

Region IIb: $\max\{\mathrm{d}(c) - |\sigma'|, 2|\sigma'|\} \leq |\sigma|$.
If $|\sigma'|$ is very small relative to $|\sigma|$ we can use the same shift sequence as in IIa, except that we replace each shift by $|\sigma'|$ with a shift by $k|\sigma'|$ for some k. This will work as long as $|\sigma| > k|\sigma'|$.

For instance let $\frac{1}{4}\mathrm{d}(c) < |\sigma'| < \frac{1}{2}|\sigma|$ and set $k = 2$. Perform the same shifts as in IIa, replacing each $|\sigma'|$ by $2|\sigma'|$. Analysis of step 2 then becomes $|\sigma| - 2|\sigma'| > |y| - \mathrm{d}(c)$ which solves to $|\sigma| > 8|\sigma'| - \mathrm{d}(c) > \mathrm{d}(c)$ since $2|\sigma'| > \frac{1}{2}\mathrm{d}(c)$.

In general when $|\sigma| > 2|\sigma'|$ we choose k such that $\frac{1}{2}\mathrm{d}(c) < k|\sigma'| < |\sigma|$ and execute the following shifts:

Start: B			**Goal:** $+|\sigma|$
No.	*Move*	*Requirement*	*Comments*
1.	$-k|\sigma'|$	$k|\sigma'| < \mathrm{d}(c)$	Choice of k above.
2.	$+|\sigma|$	$|\sigma| - k|\sigma'| < |y| - \mathrm{d}(c)$	See below
3	$+k|\sigma'|$	$|\sigma| < \mathrm{d}(c)$	Def. Repeater

For move #2 to be valid, it is necessary that $|\sigma| - k|\sigma'| < |y| - \mathrm{d}(c)$. Supposing the contrary, we get $|\sigma| - k|\sigma'| > |y| - \mathrm{d}(c) > \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma| - \mathrm{d}(c)$. Solving for $|\sigma|$ we get $|\sigma| > 4k|\sigma'| - \mathrm{d}(c)$, which implies $|\sigma| > \mathrm{d}(c)$ since $k|\sigma'| > \frac{1}{2}\mathrm{d}(c)$. Therefore move #2 is valid, which gives us a contradiction for the region.

Region III: $\max\{\frac{1}{4}\mathrm{d}(c) + \frac{5}{4}|\sigma'|, 4|\sigma'| - \mathrm{d}(c)\} \leq |\sigma| \leq 2|\sigma'|$.

Start: C			**Goal:** $-|\sigma'|$
No.	*Move*	*Requirement*	*Comments*
1.	$+|\sigma|$	$|\sigma| < \mathrm{d}(c)$	Def. Repeater
2.	$-2|\sigma'|$	$2|\sigma'| - |\sigma| < X_{\sigma'} = |y'| - \mathrm{d}(c)$	See below
3.	$-|\sigma|$	$2|\sigma'| < |y'|$	See below.
4.	$+|\sigma'|$	$|\sigma'| < \mathrm{d}(c)$	Def. Repeater

Suppose move # 2 was not valid. Then $2|\sigma'| > |\sigma| + |y'| - \mathrm{d}(c)$. Applying the repeater bound and solving for σ we get $|\sigma| < \frac{1}{4}\mathrm{d}(c) + \frac{5}{4}|\sigma'|$, contradicting the bound for Region II.

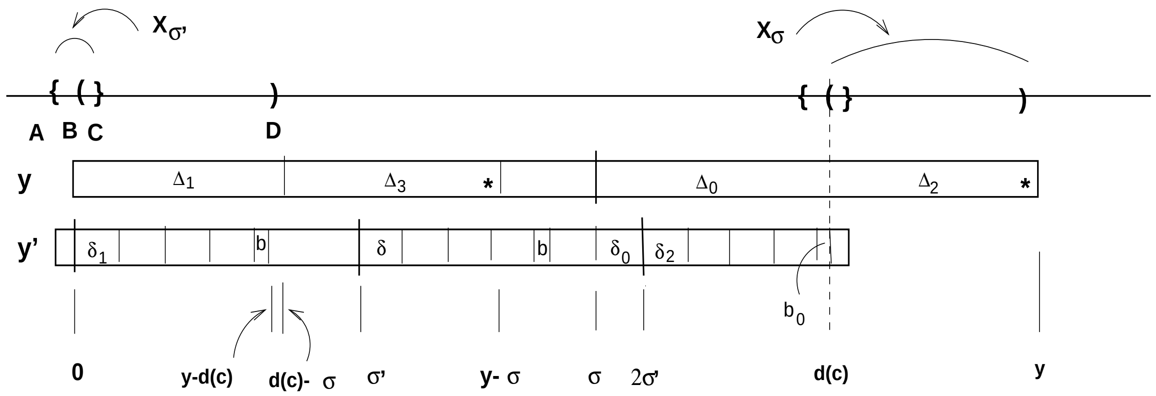

Figure 6: Proof of Lemma 3.9, Regions IV and V.

Suppose move #3 was not valid. Then applying the repeater bound and solving for $|\sigma'|$ we get $|\sigma'| > \frac{3}{5}\mathrm{d}(c)$. Applying this to our second lower bound for the region results in a contradiction:

$$|\sigma| \geq 4|\sigma'| - \mathrm{d}(c) > \frac{12}{5}\mathrm{d}(c) - \mathrm{d}(c) > \frac{7}{5}\mathrm{d}(c).$$

We have shown that all moves are valid in the region, which gives us a contradiction for the region.

We have been using three periods, $\mathrm{per}(c)$, σ and σ'. In the regions which remain, we will show that portions of $\mathrm{per}(c)$ have other periods. First we will establish that these other periods exist. As indicated in Figure 5, in the remaining regions $|\sigma'| < \frac{1}{2}\mathrm{d}(c)$.

Claim 3.11 *If* $|\sigma'| < \frac{1}{2}d(c)$,

$$d(c) - |\sigma| < |\sigma'| < |y| - |\sigma|.$$

The first inequality is true because $\mathrm{ov}(y, y') > \mathrm{d}(c)$ and we can apply Corollary 2.3. Now assume the second inequality is false, i.e. $|\sigma'| > |y| - |\sigma|$. Apply Definition 3.1 and solve for $|\sigma|$ to get $|\sigma| > 3\mathrm{d}(c) - 4|\sigma'|$. Since $|\sigma'| < \frac{1}{2}\mathrm{d}(c)$, this implies $|\sigma| > \mathrm{d}(c)$, a contradiction. This completes the proof of the claim. ∎

Figure 6 illustrates the situation, with $\mathrm{d}(c) - |\sigma| < |\sigma'| < |y| - |\sigma|$ as in the Claim above. Consider the block of characters labeled Δ_0; our notation is that the sequence of characters is labeled Δ and a particular instance of this block of characters is subscripted for reference. This block has size $|\Delta| \equiv d(c) - |\sigma|$. We can shift to the left $|\sigma|$ characters to the positions labeled Δ_1 because we remain entirely within a single copy of y when we do so. Now shift forward $\mathrm{d}(c)$ characters to Δ_2 and repeat the process. We can continue until we reach the right end of y. In the example in Figure 6, Δ_2 is not quite complete before the end of y; we have indicated with an $*$ that this block is not complete. Note that we have a region with period Δ which extends from the beginning of y to $|y| - |\sigma|$.

Now we can begin again with the block of characters labeled δ_0, which is of size $|\delta| \equiv 2|\sigma'| - |\sigma|$. We can shift left $|\sigma|$ characters to δ_1, then right $2|\sigma'|$ to δ_2, and so forth. We will repeat this process until we reach the end of $\mathrm{per}(c)$. This last partial block, labeled b_0, matches the first $|b|$ characters of block δ_0. We see that the period Δ which we first identified is composed of some number of copies of this smaller block δ with a partial block b at the end.

We can summarize this discussion by the following characterization of the periods of the cycle. There exist $x, y, z \in \mathbf{R}^+$ such that

$$\begin{aligned} \Delta &= \delta^x \\ \sigma' &= \Delta^y \\ \sigma &= \sigma'^z. \end{aligned}$$

We will now apply shift arguments to the next two regions.

Region IV: $\max\{4|\sigma'| - \mathrm{d}(c), \mathrm{d}(c) - |\sigma'| \leq |\sigma| \leq \frac{1}{2}\mathrm{d}(c) + \frac{1}{2}|\sigma'|$.

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

Requirement #2 simplifies to $2|\sigma'| < |y|$ which is certainly true since in this region $|\sigma| < \frac{1}{2}\mathrm{d}(c)$ and we know $|y| > \mathrm{d}(c)$. Negating #3 we get

$$3|\sigma'| > |y| > \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma|$$

which implies $|\sigma| < 4|\sigma'| - \mathrm{d}(c)$, which contradicts one of the lower bounds of Region IV.

The requirement for move #1 is $|y| + |\sigma| < 2\mathrm{d}(c)$. We negate this to get $|\sigma| > 2\mathrm{d}(c) - |y|$. Adding our general bound Equation 2, we get $2|\sigma| > \mathrm{d}(c) + |\sigma'|$ which contradicts the upper bound for Region IV.

Region V: $\max\{4|\sigma'| - \mathrm{d}(c), \frac{1}{2}\mathrm{d}(c) + \frac{1}{2}|\sigma'|\} \leq |\sigma| \leq \mathrm{d}(c) - \frac{1}{2}|\sigma'|$.

Start: B			**Goal:** $+\lvert\sigma\rvert$
No.	*Move*	*Requirement*	*Comments*
1.	$+\lvert\sigma'\rvert$	$\lvert\sigma'\rvert < \mathrm{d}(c)$	Def. Repeater
2.	$+\lvert\sigma\rvert$	$\lvert\sigma'\rvert + \lvert\sigma\rvert < \lvert y\rvert$	Shown above.
3.	$-\Delta$	$\lvert\sigma\rvert + \lvert\sigma'\rvert - \Delta > \lvert\sigma\rvert$	Cor. 2.3
4.	$-\lvert\sigma'\rvert$	$\lvert\sigma\rvert + \lvert\sigma'\rvert - \Delta < \mathrm{d}(c)$	See below.
5.	$+\Delta$	$\Delta < \lvert\sigma\rvert - \lvert\sigma'\rvert$	See below.

Negating the requirement for move #4 and substituting the definition of Δ gives us $|\sigma| > \mathrm{d}(c) - \frac{1}{2}|\sigma'|$, contradicting the upper bound for the Region.

Finally, negate the requirement for move #5. Again substituting the definition of Δ, this gives us $2|\sigma| - |\sigma'| < \mathrm{d}(c)$, which solves to $|\sigma| < \frac{1}{2}\mathrm{d}(c) + \frac{1}{2}|\sigma'|$, violating one of the lower bounds for the Region.

Region VI: $\max\{\mathrm{d}(c) - \frac{1}{2}|\sigma'|, 4|\sigma'| - \mathrm{d}(c)\} \leq |\sigma| \leq \frac{1}{4}\mathrm{d}(c) + \frac{5}{4}|\sigma'|$

Region VI is very tightly constrained (see Figure 5) and we will identify specific characteristics which will imply that one or more of per(c),σ, or σ' is reducible. Figure 7 depicts a representative point in this region.

We have drawn Figure 7 so that σ' contains two complete copies of Δ, and Δ contains two complete copies of δ. We now show that in Region VI, both of these facts are true.

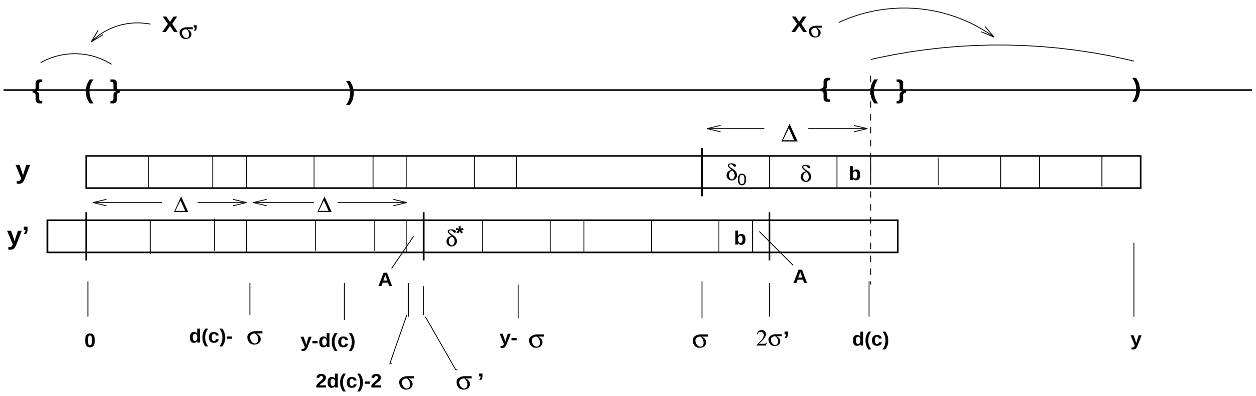

Figure 7: Proof of Lemma 3.9, Region VI.

Claim 3.12 *In Region VI,* $2\Delta < |\sigma'| < 3\Delta$*.*
Proof: For purpose of contradiction let $2(\mathrm{d}(c) - |\sigma|) > |\sigma'|$. Solving for $|\sigma|$ we have $|\sigma| < \mathrm{d}(c) - \frac{1}{2}|\sigma'|$, contradicting the first lower bound for the region. To prove the upper bound let $|\sigma'| > 3(\mathrm{d}(c) - |\sigma|)$. Solving for $|\sigma|$ we get $|\sigma| > \mathrm{d}(c) - \frac{1}{3}|\sigma'|$, which falls outside of Region VI. ■

Claim 3.13 *In Region VI,* $2\delta < d(c) - |\sigma| < 3\delta$*.*
Proof: Taking the same approach as above, negating the first inequality we get $|\sigma| < 4|\sigma'| - \mathrm{d}(c)$, contradicting the second lower bound for the Region. Negating the second inequality gives us $|\sigma| > 3|\sigma'| - \frac{1}{2}\mathrm{d}(c)$, which is not true in Region VI. ■

We now show that block δ is reducible. Note that the copy of δ marked "δ^*" is aligned with a suffix and prefix of itself in σ. By the definition of reducibility this implies that δ is reducible. This will occur as long as $|y| - |\sigma| > |\sigma'| + 2|\sigma'| - |\sigma|$. Supposing the contrary, we get

$$\begin{aligned} 3|\sigma'| &> |y| \\ &> \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma| \\ \Rightarrow |\sigma| &< 4|\sigma'| - \mathrm{d}(c), \end{aligned}$$

contradicting a lower bound for the Region. So δ is reducible, and we can further note that its period is either A or some smaller period of which A is an integral number of copies.

As drawn, δ_0 is composed of, from the right, block A which is the fractional part of Δ in σ', and b which is the fractional part of δ in Δ, as well as a suffix from another copy of δ.

We now show that block A and b are wholly contained within δ_0. We begin with the upper bound on $|\sigma|$ for the region. We use Claims 3.12 and 3.13 to dimension A and b.

$$\begin{aligned} |\sigma| &\leq \frac{1}{4}\mathrm{d}(c) + \frac{5}{4}|\sigma'| \\ \Rightarrow 4|\sigma| &\leq 5|\sigma'| + \mathrm{d}(c) \\ \Rightarrow |\sigma'| &\leq 3(2|\sigma'| - |\sigma|) + (\mathrm{d}(c) - |\sigma|) \\ \Rightarrow |\sigma'| - 2(\mathrm{d}(c) - |\sigma|) + (\mathrm{d}(c) - |\sigma|) - 2(2|\sigma'| - |\sigma|) &\leq (2|\sigma'| - |\sigma|) \\ \Rightarrow (|\sigma'| - 2\Delta) + (\Delta - 2\delta) &\leq \delta \\ \Rightarrow |A| + |b| &\leq \delta \end{aligned}$$

We have shown that A and b are wholly contained within δ_0. We previously showed that δ is reducible to A.

Per(c), as viewed in Figure 7 in terms of y', consists of some number of copies of δ, b, and A. Since δ is reducible to A, then either b must not be reducible to A or per(c) is reducible and our proof is complete. We show that b is reducible:

For strings F,G and H and integers j and k, define $H \doteq jF + kG$ to mean that string H is composed of j copies of string F and k copies of string G in some order. Also let N_σ denote the number of copies of A in σ, $N_{\sigma'}$ the number of copies of A in one copy of σ'.

Then we know that

$$\sigma \doteq N_\sigma A + 3b \qquad 2\sigma' \doteq 2N_{\sigma'}A + 4b$$

Since σ is aligned with 2 copies of σ' except for block δ_0, δ must consist of the difference between these expressions, that is

$$\delta \doteq (2N_{\sigma'} - N_\sigma)A + b.$$

But since δ is reducible to A, we know $\delta \doteq jA$ for some j. Let $\tilde{N} = 2N_{\sigma'} - N_\sigma$. Then

$$\tilde{N}A + b \doteq \delta \doteq jA \Rightarrow b \doteq (j - \tilde{N})A.$$

Therefore b is reducible to A, which implies that per(c) is reducible, which contradicts per(c) being the period of a cycle in a minimum cycle cover. This gives us our contradiction for Region VI. We have shown a contradiction for each of six regions, the union of which accounts for all possible values of $|\sigma|$ and $|\sigma'|$, which proves the lemma. ■

Note that the two lemmas above do not show that a positive and a negative characteristic are disjoint. We have not succeeded in constructing an example of this phenomenon and conjecture that it does not occur for $(\frac{3}{4}, \frac{3}{4})$-repeaters. We will show in the next section that this does not affect our algorithm or analysis.

4 The Algorithm

We now describe our algorithm, which takes advantage of the structure described in Section 3. In Section 4.2 we prove several technical lemmas which we require in the analysis of the algorithm in Section 4.3. In Section 4.4 we show that our algorithm can be implemented to run in $O(|S| + n^3)$ time.

4.1 Algorithm ShortString

Our algorithm, like the generic one described in Section 2, begins by forming a cycle cover on the distance graph, choosing a set of representatives, and then forming a second cycle cover on the representatives. We choose representatives by identifying $(\frac{3}{4}, \frac{3}{4})$-repeaters and their corresponding characteristics.

Let v and w be the representatives of two cycles in a distance graph and β the string formed by merging v and w to form $\langle v, w\rangle$ and extending it to the left to include each string in v's cycle and to the right to include each string in w's cycle. Then the extension cost $\mathrm{Ext}(v, w) = |\beta| - |\langle v, w\rangle|$. If γ is the two-cycle containing v and w, we will sometimes also write also write $\mathrm{Ext}(\gamma)$, meaning $\min(\mathrm{Ext}(v, w), \mathrm{Ext}(w, v))$. (This is the notation we introduced informally in Section 2. By the discussion in Section 2 we know that $\mathrm{Ext}(v, w) \leq \mathrm{d}(v) + \mathrm{d}(w))$. For cycles of size 3 and larger that is more than sufficient to achieve our bound. For 2-cycles we will be able to bound $\mathrm{Ext}(\gamma)$ more tightly without sacrificing the overlap achieved by the second cycle cover. The idea is that the 2-cycle gives us the freedom to choose one end or the other on which to extend cheaply.

Our algorithm, ShortString appears below. At a high level it resembles the generic algorithm, but in Step (2) we are more careful about choosing representatives and in Step (4), we are more careful about how we extend.

Algorithm ShortString

(1) Form minimum cycle cover C on distance graph
(2) Call FindReps(c) on each cycle $c \in C$ to choose representatives R
(3) Form minimum cycle cover CC on the graph induced by R
(4) Break each cycle γ in CC:
  if γ is a 2-cycle (v,t) such that $\min(\mathrm{ov}(v,t), \mathrm{ov}(t,v)) > \frac{3}{4}(\mathrm{d}(c_v) + \mathrm{d}(c_t))$
  (a) **then if** $\mathrm{Ext}(v,t) \leq \mathrm{Ext}(t,v)$
    then Discard edge (t,v) ; Extend $\langle v,t \rangle$ by $\mathrm{Ext}(v,t)$
    else Discard edge (v,t); Extend $\langle t,v \rangle$ by $\mathrm{Ext}(t,v)$
  (b) **else** discard edge of cycle γ with least overlap; Extend each vertex w by $\mathrm{d}(c_w)$
(5) Concatenate strings from (4) to form superstring α

Our procedure FindReps(c) is the key to our improved approximation bound. We begin by identifying all of the $(\frac{3}{4}, \frac{3}{4})$-repeaters of the cycle, and the corresponding major and minor characteristics. If there is a string which does not contain a $(\frac{3}{4}, \frac{3}{4})$-repeater, we choose it as the representative; such a string cannot achieve high overlap on both edges of a two-cycle by Lemma 3.5. It may be that every string in the cycle has a $(\frac{3}{4}, \frac{3}{4})$-repeater; in this case we choose a representative in two stages. In the first stage we select a set Φ of candidate strings, and in the second stage we choose from among the strings in Φ. In order to describe this portion of our algorithm we need a more precise notion the relationship between a string and the characteristics in its cycle.

Definition 4.1 Suppose a cycle c contains a characteristic X_σ and consider a string z in c. Then z *touches* σ $t_z(\sigma)$ times, where $t_z(\sigma)$ is computed as follows:

1. Each copy of X_σ which is strictly contained in z contributes 1 to $t_z(\sigma)$.

2. If X_σ is negative, then each end of z which begins within X_σ contributes 1 to $t_z(\sigma)$.

For each major anomaly X_{σ_i}, let $t^*(\sigma_i) = \max\{t_z(\sigma_i) : z \text{ is a string in } c\}$, that is, the maximum number of times any string touches σ_i. If a string z touches each characteristic X_{σ_i} exactly $t^*(\sigma_i)$ times, then z is called a *consensus candidate*. If every string in the cycle has a $(\frac{3}{4}, \frac{3}{4})$-repeater, and there is one or more consensus candidate, we place all of the consensus candidates in Φ.

The cycle may not contain any consensus candidates. It may be that we have two strings s and u such that $t_s(\sigma_i) = t^*(\sigma_i)$ but $t_s(\sigma_j) < t^*(\sigma_j)$, while $t_u(\sigma_j) = t^*(\sigma_j)$ but $t_u(\sigma_i) < t^*(\sigma_i)$. That is, s may touch X_{σ_i} one more time than u but u touches X_{σ_j} one more time than s. In this case we place in Φ all strings which begin and end within the characteristic of the largest $(\frac{3}{4}, \frac{3}{4})$-repeater.

In the second phase of representative selection we break ties among the candidate strings in Φ. We do so by examining the nested minor characteristics. Given the parse tree representation of this nesting described in Section 3, we associate a string s with a node of the tree as follows. The set of $(\frac{3}{4}, \frac{3}{4})$-repeaters in which a string is active must all be nested within each other. Thus a string defines a path in the tree. Such a path is unique because positive characteristics are disjoint by Lemma 3.9. We associate a string with the lowest node on this path.

Recall from Section 3 that there may be as many as two negative characteristics in a cycle, and that it is possible that they are not disjoint with respect to some positive characteristic(s). Suppose we have two such negative characteristics $X_{\sigma'}$ and $X_{\sigma''}$. We do not incorporate them into the parse tree. This omission will not affect the correctness of algorithm, because the parse tree only determines our choice of representative in case of ties. A string with an active repeater that has such a non-disjoint characteristic could not be involved in a tie-breaking situation because one end of it falls outside of the major repeater in question.

If we find no way to break ties even with the minor characteristics, then the strings are very closely grouped and any choice will be equally good. The pseudocode for this appears below. If a node w in a parse tree corresponds to some characteristic X_σ, we will use the notation $\sigma(w)$ to refer to σ.

Procedure FINDREPS(c)
Find all $(\frac{3}{4}, \frac{3}{4})$-repeaters and associated characteristics in c
Label major characteristics $X_{\sigma_1}, X_{\sigma_2}, \ldots, X_{\sigma_r}$ such that $|\sigma_1| > |\sigma_2| > \ldots > |\sigma_r|$
Build the parse tree for each major characteristic X_{σ_i}
if there are one or more strings without a $(\frac{3}{4}, \frac{3}{4})$-repeater
    then $r_c \leftarrow$ longest such string
    else
        if there is at least one consensus candidate
            then let X be the major characteristic in which all consensus candidates begin
            else let X be X_{σ_1}
        Let Φ be the set of strings which begin and end in X
        if $|\Phi| = 1$
            then $r_c \leftarrow a \in \Phi$
            else $r_c \leftarrow$ TIEBREAK(Φ, X)

PROCEDURETIEBREAK(Φ, X) $\{X$ is a parse tree$\}$
Let a be the leftmost string in X; let z be the rightmost string in X
$w \leftarrow \text{lca}(a, z)$ in X
if $w = a$ or $w = z$
    then return w
    else Let anc$_a$,anc$_z$ be the children of w which are ancestors of a and z respectively
        if $|\sigma(anc_a)| \geq |\sigma(anc_z)|$
            then Return a
            else Return z

In order to analyze our algorithm to obtain an approximation ratio, we require some additional technical lemmas relating the size of $(\frac{3}{4}, \frac{3}{4})$-repeaters and their characteristics within a cycle.

4.2 Technical Lemmas

The first two straightforward lemmas bound the size of a characteristic and the size of a repeater required in order for nesting to occur.

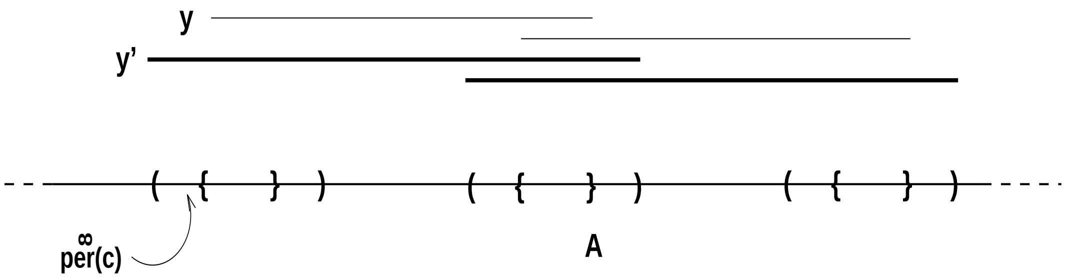

Figure 8: Proof of Lemma 4.3

Lemma 4.2 *Let X_σ be any positive characteristic in cycle c. Then $|X_\sigma| < |\sigma|$.*

Proof: Recall from Lemma 3.3 that a witness y has maximum length $|y| < \mathrm{d}(c) + |\sigma|$. So we have

$$X_\sigma = |y| - \mathrm{d}(c) < \mathrm{d}(c) + |\sigma| - \mathrm{d}(c) = |\sigma|.$$

■

Lemma 4.3 *Let X_σ be a major characteristic in cycle c and $X_{\sigma'}$ a minor characteristic nested within X_σ. Then $|\sigma| > |\sigma'|$.*

Proof: Suppose for purpose of contradiction that there is a minor characteristic X_σ nested within $X_{\sigma'}$, with $|\sigma| > |\sigma'|$. By Corollary 2.3 we must have $|\sigma| + |\sigma'| > |y| > \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}(|\sigma|)$. Solving for $|\sigma|$ gives us

$$|\sigma| > 3\mathrm{d}(c) - 4|\sigma'|. \tag{3}$$

We now apply a shift argument (see Figure 8).

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

Since only requirement #2 is conditional, we conclude its negation, $|\sigma| > |\sigma'| + |y| - \mathrm{d}(c)$. Solving for $|\sigma|$,

$$|\sigma| > 4|\sigma'| - \mathrm{d}(c). \tag{4}$$

Adding Equations 3 and 4 gives us $2|\sigma| > 2\mathrm{d}(c)$, a contradiction. ■

Lemma 4.4 *Let X_σ be a major characteristic in cycle c and $X_{\sigma'}$ a minor characteristic nested within X_σ. Then $|\sigma| \geq \frac{3}{5}d(c)$.*

Proof: In this case the witness y' is completely contained within the witness y. We apply Corollary 2.3 and the definition of $(\frac{3}{4}, \frac{3}{4})$-repeater to get

$$|\sigma| + |\sigma'| > \mathrm{ov}(y, y') > |y'| > \frac{3}{4}(\mathrm{d}(c) + |\sigma'|).$$

By Lemma 4.3 we know $|\sigma| > |\sigma'|$. Solving for $|\sigma|$ gives the desired bound. ■

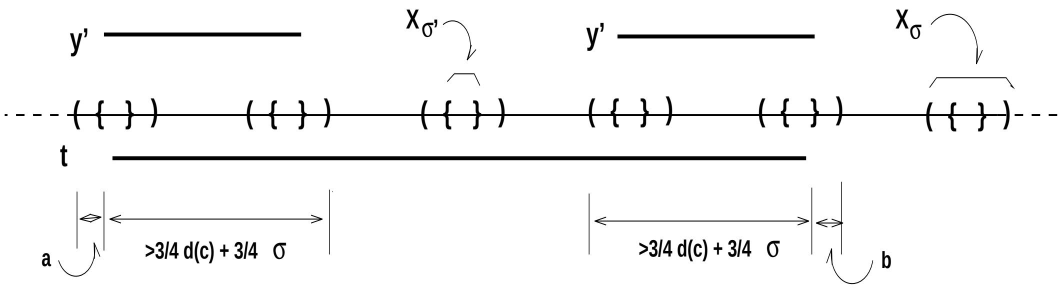

Figure 9: Proof of Lemma 4.5

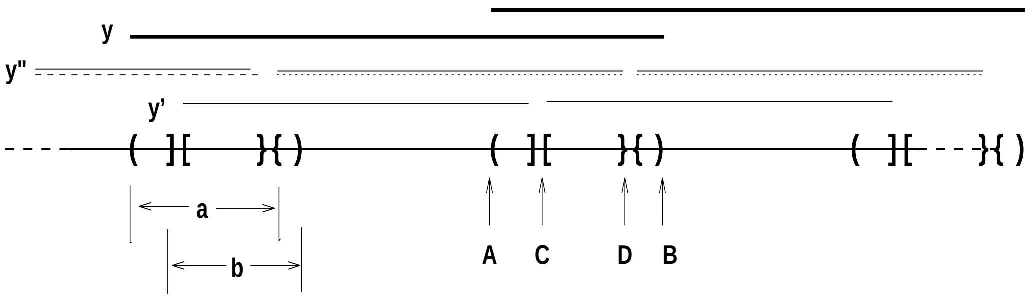

Figure 10: Proof of Lemma 4.6

Lemma 4.5 *Let X_σ be a major characteristic in cycle c and $X_{\sigma'}$ a minor characteristic nested within X_σ. If a string t has both σ and σ' active, then $d(y, y') \geq \frac{1}{2}d(c) - |\sigma'|$.*

Proof: The situation is as in Figure 9; we have dimensioned $\mathrm{d}(y, y')$ and the corresponding dimension at the right end as a and b respectively. By Corollary 2.3 we know that $|\sigma| + |\sigma'| > |y'|$.

Consider the left end of t. Since σ is active in t, the distance from the beginning of t to the rightish of the first full copy of X_σ in t must satisfy the repeater bound for σ. Since σ' is also active in t, t must have a prefix y'_ℓ which is a substring of y'. Therefore we must have $|y'| > \frac{3}{4}(\mathrm{d}(c) + |\sigma|) - b$. The same set of requirements holds for the right end of t, giving us $|y'| > \frac{3}{4}(\mathrm{d}(c) + |\sigma|) - a$. We can then solve

$$|\sigma| + |\sigma'| > |y'| > \frac{3}{4}(\mathrm{d}(c) + |\sigma|) - b$$

and the corresponding expression in a and get $a > \frac{1}{2}\mathrm{d}(c) - |\sigma'|$, $b > \frac{1}{2}\mathrm{d}(c) - |\sigma'|$. ■

Lemma 4.6 *If two minor characteristics $X_{\sigma'}$ and $X_{\sigma''}$ are nested separately within another characteristic X_σ, then $\max\{|\sigma'|, |\sigma''|\} \geq \frac{1}{3}d(c)$.*

Proof: Figure 10 depicts the situation. We show $X_{\sigma'}$ and $X_{\sigma''}$ as negative characteristics; this case gives our proof the most difficulty. All of the bounds derived in this proof apply to the case of positive characteristics, and there are additional shift techniques available in that case as well that could simplify the proof.

Case 1: $\sigma' + \sigma'' \geq \frac{2}{3}\mathrm{d}(c)$.

In this case the Lemma follows immediately.

Case 2: $\sigma' + \sigma'' < \frac{2}{3}\mathrm{d}(c)$.

We can apply Corollary 2.3 to the overlap between y' and y'' as follows:

$$|\sigma'| + |\sigma''| > \mathrm{ov}(y', y'') = y - a - b \tag{5}$$

Equation 5 and the condition for Case 2 imply that $|y| - a - b < \frac{2}{3}\mathrm{d}(c)$, or

$$a + b > y - \frac{2}{3}\mathrm{d}(c). \tag{6}$$

Since $a < |y| - \mathrm{d}(c)$ and $b < |y| - \mathrm{d}(c)$, we know

$$|y| > \frac{4}{3}\mathrm{d}(c) \tag{7}$$

We will now use two shift techniques as we did in proving Lemma 3.9. Rather than using different techniques for different values of our parameters, we will derive inequalities from each set of shifts which we will combine to complete the proof.

The first shift technique utilizes moves from positions A and B (see Figure 10). We show those from A; those from B are directly analogous.

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

Negating the requirement for move #1, and using Equation 5 as a lower bound on b, we have

$$\begin{aligned}
|\sigma'| &> 2\mathrm{d}(c) - |y| + b - (\mathrm{d}(c) - |y'|) \\
&> \mathrm{d}(c) + |y'| - |y| + (|y| - a - |\sigma'| - |\sigma''|) \\
\Rightarrow \frac{5}{4}|\sigma'| + |\sigma''| &> \frac{7}{4}\mathrm{d}(c) - a \\
&> \frac{3}{4}\mathrm{d}(c)
\end{aligned}$$

which implies $|\sigma'| + |\sigma''| > \frac{2}{3}\mathrm{d}(c)$, contradicting the bound for Case 2. So move #1 is legal.

We next show that the third move is legal, which will allow us to infer bounds based on the requirement for move #2. Negating the requirement for move #3, and using Equation 5 as a lower bound on b, we have

$$\begin{aligned}
|\sigma| - |\sigma'| &< |y| - |y'| - b \\
&< |y| - |y'| - (|y| - a - |\sigma'| - |\sigma''|) \\
\Rightarrow 2|\sigma'| + |\sigma''| &> |\sigma| + |y'| - a \\
\Rightarrow \frac{5}{4}|\sigma'| + |\sigma''| &> |\sigma| + \frac{3}{4}\mathrm{d}(c) - |\sigma| \\
\Rightarrow \frac{9}{4}\max\{|\sigma'|, |\sigma''|\} &> \frac{3}{4}\mathrm{d}(c) \\
\Rightarrow \max\{|\sigma'|, |\sigma''|\} &> \frac{1}{3}\mathrm{d}(c)
\end{aligned}$$

The only condition which remains is the requirement for move #2, that $|\sigma| - |\sigma'| < |y| - \mathrm{d}(c)$. Since there is only one condition which could possibly be false to avert a contradiction, we may conclude that it is false:

$$|\sigma| - |\sigma'| > |y| - \mathrm{d}(c). \tag{8}$$

The corresponding set of moves from B is $+|\sigma''| - |\sigma| - |\sigma''|$ and the same analysis allows us to conclude

$$|\sigma| - |\sigma''| > |y| - \mathrm{d}(c). \tag{9}$$

We now employ another shift technique, this time from positions C and D. Again we show only the shifts from C; those from D are analogous.

Start: C			**Goal:** $+|\sigma'|$
No.	*Move*	*Requirement*	*Comments*
1.	$-|\sigma|$	$|\sigma'| < \mathrm{d}(c)$	Def. repeater.
2.	$+|\sigma'|$	$|\sigma| - |\sigma'| < \mathrm{d}(c) - |y'|$	See below
3	$+|\sigma|$	$|\sigma'| < |y'| + b - \mathrm{d}(c)$	See below.

Move #2 requires $|\sigma| - |\sigma'| > \mathrm{d}(c) - |y'|$. This must be true since $|y'| > \frac{3}{4}\mathrm{d}(c)$ (definition of repeater), $|y| > \frac{4}{3}\mathrm{d}(c)$ (Equation 7), and $|\sigma| - |\sigma'| > |y| - \mathrm{d}(c)$ (the shift bound in equation 8).

The only requirement which might not be true is #3, so we conclude its negation

$$|\sigma'| > |y'| + b - \mathrm{d}(c). \tag{10}$$

The corresponding set of moves from D gives

$$|\sigma''| > |y''| + a - \mathrm{d}(c) \tag{11}$$

We add Equations 10 and 11 and use lower bounds on $|y|$ and $a + b$ from Equations 6 and 7 to get

$$\begin{aligned}
|\sigma'| + |\sigma''| &> |y'| + |y''| + a + b - 2\mathrm{d}(c) && (12)\\
&> \frac{3}{2}\mathrm{d}(c) + \frac{3}{4}(|\sigma'| + |\sigma''|) + |y| - \frac{2}{3}\mathrm{d}(c) - 2\mathrm{d}(c) && (13)\\
\frac{1}{4}|\sigma'| + |\sigma''| &> (\frac{3}{2} + \frac{4}{3} - \frac{8}{3})\mathrm{d}(c) && (14)\\
|\sigma'| + |\sigma''| &> \frac{2}{3}\mathrm{d}(c) && (15)
\end{aligned}$$

■

Lemma 4.7 *A cycle c can contain no more than one $(\frac{3}{4}, \frac{3}{4})$-repeater with $|\sigma| = k$, $1 \le k \le d(c)$.*

Proof: Suppose for purpose of contradiction that there are two such $(\frac{3}{4}, \frac{3}{4})$-repeaters with $|\sigma| = |\sigma'|$. Their witnesses can be offset by at most $\frac{1}{2}\mathrm{d}(c)$, so

$$\mathrm{ov}(y, y') \ge |y| - \frac{1}{2}\mathrm{d}(c) \ge \frac{3}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma| - \frac{1}{2}\mathrm{d}(c) = \frac{1}{4}\mathrm{d}(c) + \frac{3}{4}|\sigma| \ge |\sigma|.$$

But if they overlap by more than $|\sigma|$ then $\sigma = \sigma'$, a contradiction. ■

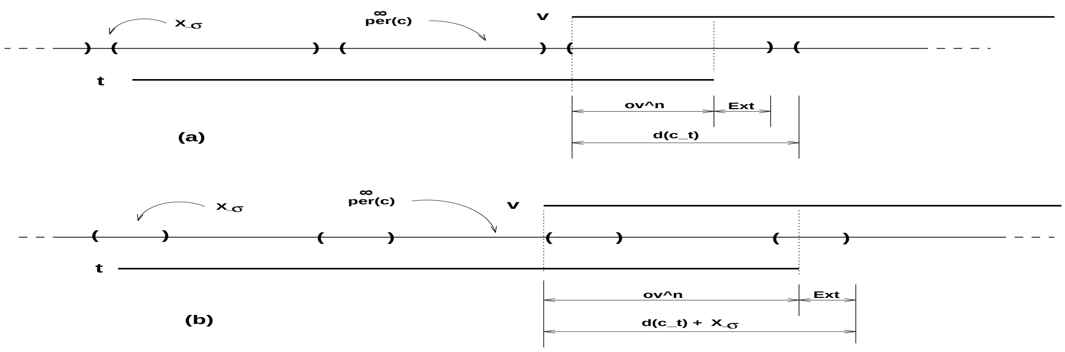

Figure 11: Cases 1a and 1b from Proof of Lemma 4.9

4.3 The Approximation Ratio

We now analyze the algorithm. Recall from Section 2 that given a 2-cycle γ in CC composed of vertices v and t, we use ov^n_γ to denote the overlap on the edge not taken and ov^t_γ to denote the overlap taken. Recall also from Lemma 2.2 that in order to improve the approximation bound from 3 down to $1+\beta$ for $\beta \geq \frac{5}{3}$, we need only to show that for all 2-cycles γ, $\mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) \leq \beta(\mathrm{d}(c_v) + \mathrm{d}(c_t))$.

If $\min(\mathrm{ov}(v,t), \mathrm{ov}(t,v)) \leq \frac{3}{4}(\mathrm{d}(c_v) + \mathrm{d}(c_t))$ then $\mathrm{ov}^n_\gamma \leq \frac{3}{4}(\mathrm{d}(c_v) + \mathrm{d}(c_t))$ and we can achieve the bound of $\frac{7}{4}(\mathrm{d}(c_v) + \mathrm{d}(c_t)$ with an extension cost of $\mathrm{d}(c_v) + \mathrm{d}(c_t)$. Either our algorithm succeeded in finding a representative with no $(\frac{3}{4}, \frac{3}{4})$-repeaters, or there was not an r_v available such that $\mathrm{per}(v) = \sigma$ for any $(\frac{3}{4}, \frac{3}{4})$-repeater in t. Thus we have

Lemma 4.8 *Let γ be a 2-cycle in CC with v the representative of cycle c_v and t the representative of c_t, and let $min(ov(v,t), ov(t,v)) < \frac{3}{4}(d(c_v) + d(c_t))$. Then $ov^n_\gamma + Ext(\gamma) \leq \frac{7}{4}(d(c_v) + d(c_t))$.*

When there is high overlap on both edges of a 2-cycle, we can achieve the same bound due to our choice of representative.

Lemma 4.9 *Let γ be a 2-cycle in CC with v the representative of cycle c_v and t the representative of c_t, and let $min(ov(v,t), ov(t,v)) \geq \frac{3}{4}(d(c_v) + d(c_t))$. Then $ov^n_\gamma + Ext(\gamma) \leq \frac{7}{4}(d(c_v) + d(c_t))$.*

Proof: Assume without loss of generality $\mathrm{d}(c_t) \geq \mathrm{d}(c_v)$. Since v achieves high overlap at both ends of t, by Lemma 3.5 there must be at least one $(\frac{3}{4}, \frac{3}{4})$-repeater σ' in c_t, and in fact $\sigma' = \mathrm{per}(v)$. Furthermore, we observe that all strings in c_t must have at least one $(\frac{3}{4}, \frac{3}{4})$-repeater; if some string t' did not, we would have chosen it as the representative of c_t and this high-overlap two-cycle could not have occurred.

We consider three cases which depend on the number and size of repeaters found in FINDREPS(c_t):

1. One repeater.
2. Two repeaters $|\sigma'| < |\sigma| < \frac{3}{5}\mathrm{d}(c_t)$.
3. At least two repeaters, with larger repeater $|\sigma| > \frac{3}{5}\mathrm{d}(c_t)$

Case 1. Since we only found one repeater σ in FINDREPS(c_t), and t then appeared in a high-overlap two-cycle with a vertex v, then it must be the case that $\mathrm{per}(v) = \sigma$. Furthermore, since there was only one repeater t must have been chosen as a consensus candidate.

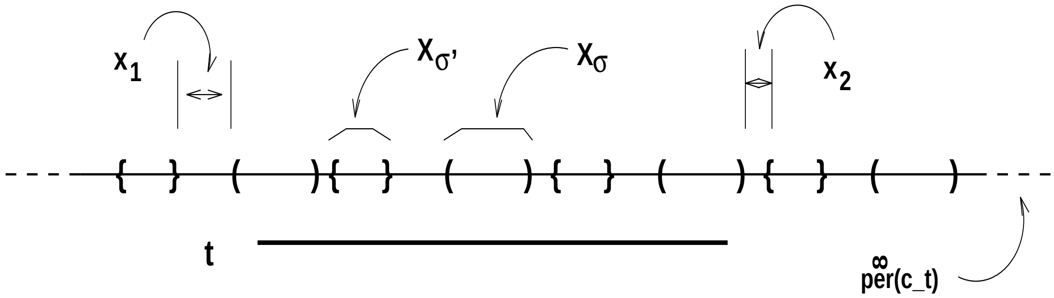

Figure 12: Case 2 from Proof of Lemma 4.9

To extend t to include another string t' in c_t, we align the copies of X_σ in t' with those in t. Figures 11a and 11b illustrate this situation when σ has, respectively, a negative and positive characteristic. The figures also suggest the tradeoff at work in the analysis; recall that ov^n_γ is the edge of the 2-cycle in the second cycle cover CC which is not used in forming our superstring. As the figures suggest, as ov^n_γ increases, the remaining portion of the cycle to be covered by extension decreases. This is not the case with an arbitrary choice of representative, but only because we have identified the structure of the strings in c. We bound the overlap and extension costs for the case of a negative and for a positive characteristic.

Case 1a. Since t was a consensus candidate, no string t' can touch X_σ more times than t. Our extension cost therefore cannot exceed $\mathrm{d}(c_t) - \mathrm{ov}^n_\gamma$ as dimensioned in the figure. Allowing as well for extending v to cover the remaining strings in c_v at the cost of $\mathrm{d}(c_v) = |\sigma|$ we have

$$\begin{aligned} \mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}^n_\gamma + \mathrm{d}(c_t) - ov^n_\gamma + \mathrm{d}(c_v) \\ &= \mathrm{d}(c_t) + \mathrm{d}(c_v). \end{aligned}$$

Case 1b. Now X_σ is a positive characteristic. However, since t was chosen as a consensus candidate no string t' can extend beyond X_σ. This amount of extension, as dimensioned in Figure 11b, is $\mathrm{d}(c_t) + |X_\sigma| - ov^n_\gamma$. Again we allow for extending by $\mathrm{d}(c_v) = |\sigma|$, giving us

$$\begin{aligned} \mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}^n_\gamma + \mathrm{d}(c_t) + |X_\sigma| - \mathrm{ov}^n_\gamma + \mathrm{d}(c_v) \\ &\leq \mathrm{d}(c_t) + 2\mathrm{d}(c_v) \\ &\leq \frac{3}{2}(\mathrm{d}(c_t) + \mathrm{d}(c_v)). \end{aligned}$$

Case 2. Suppose we had two $(\frac{3}{4}, \frac{3}{4})$-repeaters in c_t with $|\sigma'| < |\sigma| < \frac{3}{5}\mathrm{d}(c_t)$. By Lemma 4.4 neither can be nested within the other. If in $\textsc{FindReps}(c_t)$ we chose t because it was a consensus candidate, that is $t_t(\sigma) = t^*(\sigma)$ and $t_t(\sigma') = t^*(\sigma')$, then no other string t' could touch another copy of either characteristic, and the analysis would proceed as in Case 1 where we have only one $(\frac{3}{4}, \frac{3}{4})$-repeater.

Suppose there was not such a consensus candidate, so t was chosen because it has σ, the largest $(\frac{3}{4}, \frac{3}{4})$-repeater as a $(\frac{3}{4}, \frac{3}{4})$-repeater. Let x_1 and x_2 be the two distances between X_σ and $X_{\sigma'}$. Let $b = \max\{x_1, x_2\} \geq \frac{1}{2}(x_1 + x_2) \geq \frac{1}{2}(\mathrm{d}(c_t) - |\sigma| - |\sigma'|)$. Then we extend in the less expensive direction, giving us

$$\begin{aligned} \mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}^n_\gamma + 2\mathrm{d}(c_t) - b - \mathrm{ov}^n_\gamma + \mathrm{d}(c_v) \\ &\leq \frac{3}{2}\mathrm{d}(c_t) + \frac{1}{2}(|\sigma| + |\sigma'|) + \mathrm{d}(c_v) \end{aligned}$$

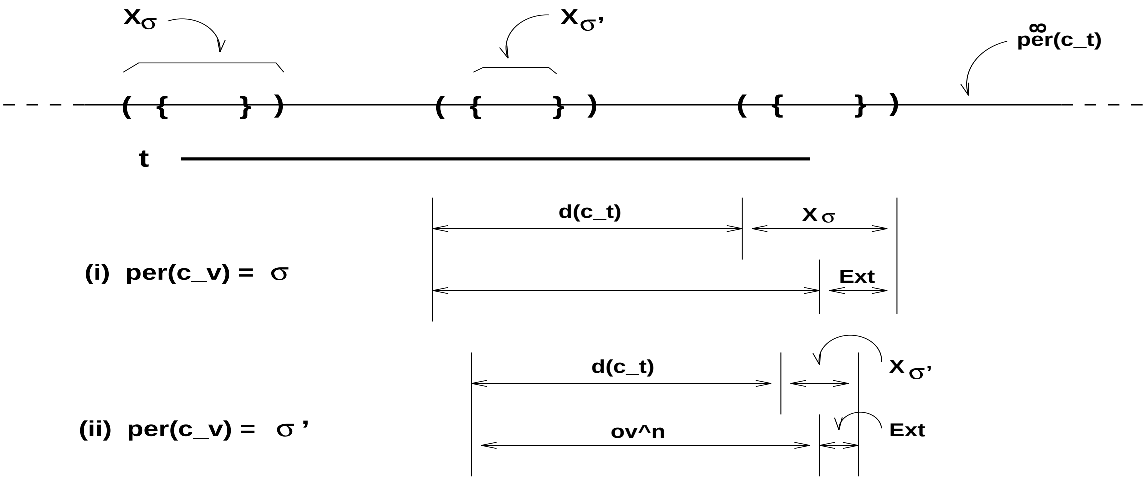

Figure 13: Proof of Lemma 4.9,Case 3A

$$
\begin{aligned}
&\leq \frac{3}{2}\mathrm{d}(c_t) + 2\mathrm{d}(c_v) \\
&\leq \frac{27}{16}(\mathrm{d}(c_t) + \mathrm{d}(c_v)).
\end{aligned}
$$

This last follows from the fact that $\sigma < \frac{3}{5}\mathrm{d}(c)$.

Case 3. Recall that in this case we have at least two repeaters and the larger repeater $|\sigma| > \frac{3}{5}\mathrm{d}(c_t)$. Here we may have several major characteristics, and we may also have minor characteristics nested according to the constraints of Lemmas 3.8, 3.9, 4.4, and 4.5.

There are two ways that t could have been chosen as a representative; either it was a consensus candidate, or it had the largest $(\frac{3}{4}, \frac{3}{4})$-repeater σ. There are two ways in which ties might be broken by TieBreak. If one candidate was an ancestor of the other in the parse tree representation, we will say that the tie was broken *strongly*; otherwise, each had an ancestor that were siblings, and we say that the tie was broken *weakly*. We consider each of the resulting four cases:

Subcase A. Suppose t was a consensus candidate, and any ties were broken strongly. Let X_σ be the major characteristic in which t begins and ends as in Figure 13. If $\mathrm{per}(v) = \sigma$, and using Lemma 4.2, then

$$
\begin{aligned}
\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + \mathrm{d}(c_t) + |X_\sigma| - \mathrm{ov}_\gamma^n + \mathrm{d}(c_v) \\
&\leq \mathrm{d}(c_t) + 2\mathrm{d}(c_v) \\
&\leq \frac{3}{2}(\mathrm{d}(c_t) + \mathrm{d}(c_v)).
\end{aligned}
$$

It could be that $\mathrm{per}(v) = \sigma'$, with $X_{\sigma'}$ nested within X_σ. If the strings in c_t were grouped so that there were no ties that could be broken, then no string extends beyond $X_{\sigma'}$.

$$
\begin{aligned}
\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + \mathrm{d}(c_t) + |X_{\sigma'}| - \mathrm{ov}_\gamma^n + \mathrm{d}(c_v) \\
&\leq \mathrm{d}(c_t) + 2\mathrm{d}(c_v) \\
&\leq \frac{3}{2}(\mathrm{d}(c_t) + \mathrm{d}(c_v))
\end{aligned}
$$

If a tie was broken, then it was broken strongly, so σ' had some characteristic nested within it. Then by Lemma 4.4 $|\sigma'| > \frac{3}{5}\mathrm{d}(c)$, giving us

$$\begin{aligned} \mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + \mathrm{d}(c_t) + |X''_\sigma| - \mathrm{ov}_\gamma^n + \mathrm{d}(c_v) \\ &\leq 2\mathrm{d}(c_t) + \mathrm{d}(c_v) \\ &\leq \frac{13}{8}(\mathrm{d}(c_t) + \mathrm{d}(c_v)) \end{aligned}$$

<u>Subcase B.</u> In this case t was chosen because it had the largest $(\frac{3}{4}, \frac{3}{4})$-repeater σ, and any ties were broken strongly. No other string t' in c can touch the last major characteristic before the next copy of X_σ or such a string would have had the maximum number of touches of each major characteristic and would have been made the representative. First suppose that $\mathrm{per}(v) = \sigma$. Because we're in Case 3, we know that $|\sigma| \geq \frac{3}{5}\mathrm{d}(c_t)$ and

$$\begin{aligned} \mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + 2\mathrm{d}(c_t) - \mathrm{ov}_\gamma^n + \mathrm{d}(c_v) \\ &\leq 2\mathrm{d}(c_t) + \mathrm{d}(c_v) \\ &\leq \frac{13}{8}(\mathrm{d}(c_t) + \mathrm{d}(c_v)) \end{aligned}$$

It could also be that $\mathrm{per}(v) = \sigma'$, with $X_{\sigma'}$ nested within X_σ. If there was a tie broken strongly, then $|\sigma'| > \frac{3}{5}\mathrm{d}(c_t)$ and the analysis is as above. Similarly if there was no tie break but $|\sigma'| > \frac{3}{5}\mathrm{d}(c_t)$ then the same analysis as above gives us a bound of $\frac{5}{3}(\mathrm{d}(c_t) + \mathrm{d}(c_v))$. If there was no tie broken, and $|\sigma'| < \frac{3}{5}\mathrm{d}(c_t)$ then t must have had both σ and σ' active and we apply Lemma 4.5:

$$\begin{aligned} \mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + 2\mathrm{d}(c_t) - (\frac{1}{2}\mathrm{d}(c_t) - |\sigma'|) - \mathrm{ov}_\gamma^n + \mathrm{d}(c_v) \\ &\leq \frac{3}{2}\mathrm{d}(c_t) + 2\mathrm{d}(c_v) \\ &\leq \frac{27}{16}(\mathrm{d}(c_t) + \mathrm{d}(c_v)). \end{aligned}$$

<u>Subcase C.</u> Now suppose t was a consensus candidate, and there was a tie which was broken weakly. Suppose t started and ended in major characteristic X_σ. Then any other string t' in the cycle can be aligned with t and not extend beyond the end of X_σ. We know that $|\sigma| > \frac{3}{5}\mathrm{d}(c_t)$ by Lemma 4.4. We might have $\mathrm{per}(c_v) = \sigma$ or $\mathrm{per}(c_v) = \sigma'$ for some σ' nested in σ. If $\mathrm{per}(c_v) = \sigma$, then the analysis follows that of Subcase A when we get $\mathrm{per}(c_v) = \sigma$, and our bound is $\frac{3}{2}\mathrm{d}(c)$.

Now suppose $\mathrm{per}(c_v) = \sigma'$ for some σ' nested within σ. Then since $|\sigma'| > \frac{1}{3}\mathrm{d}(c)$ by Lemma 4.6,

$$\begin{aligned} \mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + \mathrm{d}(c_t) + |X_\sigma| - \mathrm{ov}_\gamma^n + \mathrm{d}(c_v) \\ &\leq 2\mathrm{d}(c_t) + \mathrm{d}(c_v) \\ &\leq \frac{7}{4}(\mathrm{d}(c_t) + \mathrm{d}(c_v)). \end{aligned}$$

<u>Subcase D.</u> Finally, consider the case when t was chosen because it had the largest $(\frac{3}{4}, \frac{3}{4})$-repeater σ, and there was a tie broken weakly. As in Subcase B, no other string t' can touch the last major characteristic before the next copy of X_σ, or it would have been chosen representative on the basis of maximum touches. If we get $\mathrm{per}(c) = \sigma$ then the analysis will follow that of Subcase B, and we

will get a bound of $\frac{13}{8}(\mathrm{d}(c_t) + \mathrm{d}(c_v))$. Finally suppose we get $\mathrm{per}(c_v) = \sigma'$, for some σ' nested inside of σ. Then again using Lemma 4.6,

$$\begin{aligned} \mathrm{ov}^n_\gamma + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}^n_\gamma + 2\mathrm{d}(c_t) - \mathrm{ov}^n_\gamma + \mathrm{d}(c_v) \\ &\leq 2\mathrm{d}(c_t) + \mathrm{d}(c_v) \\ &\leq \frac{7}{4}(\mathrm{d}(c_t) + \mathrm{d}(c_v)). \end{aligned}$$

This completes the four subcases of Case 3, which completes the proof of the Lemma. ■

We now combine Lemmas 2.2, 4.8, and 4.9 with the results of Section 3 to obtain:

Theorem 4.10 *Algorithm ShortString is a $2\frac{3}{4}$-approximation for the shortest superstring problem.*

4.4 Analysis of Running Time

The running time of our algorithm is dominated by three tasks: distance graph construction, the computation of the two cycle covers, and choosing representatives.

Distance graph construction. The first step in our algorithm is to build the distance graph G from the set S of n strings. This requires finding the maximal overlap between each ordered pair of strings. A naive algorithm for this task requires time $\Theta(n|S|)$. However, Gusfield et al.[8] have given a solution to this problem which runs in time $\Theta(|S| + n^2)$. This is optimal because the first term is required simply to read the input, and the second term is needed to write the output.

Cycle cover computation. Our algorithm requires two cycle cover computations. The well-known solution to this problem is to reduce it to an assignment problem, which can be solved in $O(n^3)$ [12].

Choosing representatives. The critical task of our procedure FindReps(c) is to identify the $(\frac{3}{4}, \frac{3}{4})$-repeaters of each cycle. Once this has been done, it remains to compute the number of touches $t_z(\sigma_i)$ for each string z and each repeater σ_i, and to call TieBreak(Φ, X) to choose from among the candidates in Φ. Computing the $t_z(\sigma_i)$ for the strings in a cycle c can be done in a straightforward way in $O(\mathrm{d}(c))$, which sums over all of the cycles to $O(\mathrm{d}(C)) = O(|S|)$. TieBreak($\Phi, X$) runs in $O(h)$ where h is the height of the parse tree rooted at major characteristic X. Clearly $h = O(\mathrm{d}(c))$, so this step summed over all cycles is also $O(|S|)$.

It remains to determine how much time is required to find the $(\frac{3}{4}, \frac{3}{4})$-repeaters in a cycle. A repeater by definition must be active in at least one string, so we can limit ourselves to looking for $(\frac{3}{4}, \frac{3}{4})$-repeaters which are witnessed by a prefix and suffix y_ℓ and y_r of some string rather than arbitrary points in $\mathrm{per}(c)$. In particular we note that the characteristics of all repeaters active for a given string must be nested. For each string z, we will first examine it for large repeaters ($|\sigma| > \frac{1}{3}\mathrm{d}(c)$) and then for small ones. We describe our method for finding large repeaters, followed by a remark on the slight alteration required for small repeaters.

We outline our method for finding large repeaters as follows:

1. Find a set L of candidate pairs $(\sigma_i, y_\ell(\sigma_i))$.

2. Find a set R of candidate pairs $(\sigma_i, y_r(\sigma_i))$.

3. Remove from L and R candidate pairs which do not satisfy the $(\frac{3}{4}, \frac{3}{4})$-repeater bound.

4. $L \cap R$ is the set of $(\frac{3}{4}, \frac{3}{4})$-repeaters.

Steps 3 requires testing each candidate pair for whether it meets the repeater bound. Each test is $O(1)$, and there are $O(|s|)$ of them. Step 4, computing the intersection of L and R, is also straightforward. By Lemma 4.7 $|\sigma_i|$ uniquely determines σ_i, and all of the σ_i are ordered, so computing the intersection is a simple $O(\mathrm{d}(c))$ operation. Steps 1 and 2 are symmetric; without loss of generality we describe Step 1. We will examine a series of candidates for y_ℓ. This series will be chosen in such a way that it will be sufficient to determine the smallest period of each candidate. That is, given a candidate witness y_ℓ, we wish to determine the smallest σ_i such that $y_\ell = (\sigma_i)^{x_i}$ for some $x_i \in \mathbf{R}^+$.

We will utilize the pre-processing step of the string-matching algorithm due to Knuth, Morris and Pratt [9], which we will refer to as KMP-P. In what follows, let $s[1..j]$ denote the prefix of length j of some string s. The subroutine KMP-P accepts as input a string z of length k. It computes for each position i in z a value $\pi[i] = \mathrm{ov}(z[1..i], z)$; that is, the length of the longest suffix of $z[1..i]$ which is also a prefix of z. Note that $\pi[k] = \mathrm{ov}(z, z)$. As was observed by Gusfield in [7], this self-overlap implies that z has period $|\sigma_1| = k - \pi[k]$; since $\pi[k]$ is the largest self-overlap, σ_1 is the smallest period of z. This subroutine runs in time $O(|z|)$.

The range of π can be partitioned into subranges; each subrange consists of a sequence which increases by exactly 1. Each of these subranges corresponds to a single alignment of z with itself. We augment KMP-P to compute these subranges. In particular for any index i we wish to know the end of the subrange containing i; that is, let $\psi[i]$ be the smallest index $j > i$ such that $\pi[j] \geq \pi[j+1]$. We can compute ψ with an additional pass over π.

Let the *modulus* of string s be given by $m_s = |s| \,\%\, \mathrm{d}(c)$. If z has at least one large $(\frac{3}{4}, \frac{3}{4})$-repeater, let σ_1 be the smallest such $(\frac{3}{4}, \frac{3}{4})$-repeater. We observe that $z[1..m_z + \mathrm{d}(c)]$ must be a substring of $y_\ell(\sigma_1)$; that is, it must be periodic in σ_1.

In order to find set L we first run KMP-P on z, computing π and ψ. Now let $h_0 = m_z + \mathrm{d}(c)$. Then $|\sigma_1| = h_0 - \pi[h_0]$, with σ_1 itself given by the first $|\sigma_1|$ characters of z. Now we find the rightish of $y_\ell(\sigma_1)$ by finding the next place where the sequence of π values is broken, that is $\psi(h_0)$. Any choice of h_0 up to and including $\psi(h_0)$ would have given us the same σ_1, and any $h_0 > \psi(h_0)$ would have given us some $|\sigma'| > |\sigma_1|$. We have now found the first candidate pair $(\sigma_1, y_\ell(\sigma_1))$.

We observed above that the characteristics of all $(\frac{3}{4}, \frac{3}{4})$-repeaters which are active in z must be nested. By Lemma 4.3, we know that if z has two $(\frac{3}{4}, \frac{3}{4})$-repeaters with X_{σ_1} nested within X_{σ_2}, then $|\sigma_2| > |\sigma_1|$. Therefore the rightish at the end of $y_\ell(\sigma_2)$ will be strictly to the right of that of $y_\ell(\sigma_1)$. To find σ_2, let $h_1 = \psi(h_0) + 1$. We get $|\sigma_2| = h_1 - \pi[h_1]$, and the rightish of $y_\ell(\sigma_2)$ is at $\psi(h_1)$.

In general, $|\sigma_{i+1}| = h_i - \pi[h_i]$, $|y_\ell(\sigma_i)| = \psi(h_i)$, and $h_{i+1} = \psi[h_i]$. We can terminate when $|\sigma_{i+1}| \geq \mathrm{d}(c)$.

To find small repeaters, we repeat the procedure outline above using $h_0 = m_z$ and terminating when $|\sigma_{i+1}| \geq \frac{1}{3}\mathrm{d}(c)$.

Since we can find all of the $(\frac{3}{4}, \frac{3}{4})$-repeaters in any string s in $O(|s|)$, we can find all of the $(\frac{3}{4}, \frac{3}{4})$-repeaters in all of the strings in $O(|S|)$. Thus we can choose representatives in $O(|S|)$, which matches the time required for the construction of the distance graph. The cycle cover computations require an additive $O(n^3)$, which gives us a total running time of $O(|S| + n^3)$.

5 Discussion

We believe that the machinery presented here will ultimately lead to better approximation bounds than $2\frac{3}{4}$ for the shortest superstring problem. We know more about the structure than is included in this extended abstract. We have studied the properties of $(\frac{2}{3}, \frac{2}{3})$-repeaters and believe that they can be used to achieve a bound of $2\frac{2}{3}$. We further conjecture that, by combining the structure of high-overlap 2-cycles with some recent maximum TSP work [10], we may achieve improved bounds. We also hope that the understanding we have developed about the particular structure of strings will help in solving real biological problems, which may have similar structure.

Acknowledgements

We thank Rao Kosaraju for pointing out a flaw in an earlier proof of this result, Shanghua Teng for helpful discussions and for sharing his unpublished work on superstrings, James Park for many helpful discussions and reading a draft of this paper, Bill Smyth for helpful discussions, and Perry Fizzano for reading a draft of this paper and for preparing some of the figures.

References

[1] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis. Linear approximation of shortest superstrings. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 328–336, 1991. To appear in JACM.

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

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

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

[5] 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.

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

[7] D. Gusfield. Faster implementation of a shortest superstring approximation. *Information Processing Letters*, (51):271–274, 1994.

[8] 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.

[9] D.E. Knuth, J.H.Morris, and V.B. Pratt. Fast pattern matching in strings. *SIAM Journal on Computing*, 6:189–195, 1977.

[10] 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.

[11] 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.

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

[13] 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.

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

[15] J. Tarhio and E. Ukkonen. A greedy approximation algorithm for constructing shortest common superstrings. *Theoretical Computer Science*, 57:131–145, 1988.

[16] 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.

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