7-14-1994

A 2-3/4-Approximation Algorithm for the Shortest Superstring Problem

Chris Armen
Dartmouth College

Clifford Stein
Dartmouth College

A 2 3/4-APPROXIMATION ALGORITHM FOR THE SHORTEST SUPERSTRING PROBLEM

Chris Armen
Clifford Stein

Technical Report PCS-TR94-214
Revised 10/4/94

10/94

A $2\frac{3}{4}$-Approximation Algorithm for the Shortest Superstring Problem

(extended abstract)

Chris Armen* Clifford Stein †

Department of Computer Science
Dartmouth College
Hanover, NH

July 14, 1994

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), [13] (factor of $2\frac{8}{9}$), [2] (factor of $2\frac{5}{6}$) and [7] (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. 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 and data compression. In some DNA sequencing problems, there is a long string of DNA that one wishes to identify, but current technology only enables one to identify short strings with any confidence. Hence, the current practice is to obtain a collection of short strings and then attempt to reconstruct the larger string. An algorithm for the shortest superstring problem is an important component in this process [3, 8, 10]. A shortest superstring can also be used to perform data compression on strings with large amounts of similarity by representing them as a shortest superstring α and indices into α [6, 11].

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 [14, 12] that repeatedly merges the pair of strings with the maximum amount of overlap. The first provable bound on the performance 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 [13] 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 [7].

The algorithms in [13, 2, 7] 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 shorter 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. (As we shall see, non-periodic strings are not really of interest.) 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 polynomial time. As stated, it has to spend more time than previous $O(1)$-approximation algorithms looking at the characters in the strings. The running time can be decreased by using appropriate data structures, but we do not address that issue in this extended abstract.

2 Preliminaries

For consistency, we use some notation and definitions of [1] and [13]. 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 $t = xv$. We call u the *prefix* of s with respect to t, $\text{pref}(s,t)$, and refer to $|u|$ as the distance from s to t, $d(s,t)$. Observe that for any s and t, $\text{ov}(s,t) + d(s,t) = |s|$. String uxv, the shortest superstring of s and t in which s appears before t is denoted by $\langle s,t \rangle$, and $|\langle s,t \rangle| = |s| + |t| - \text{ov}(s,t)$.

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 $d(s_i,s_j)$ and label the edge with $\text{pref}(s,t)$. 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 $\text{pref}(s_{\pi_i}, s_{\pi_j})$, almost corresponds to a superstring in S, the only difference being that we must replace $\text{pref}(s_{\pi_n}, s_{\pi_1})$ with s_{π_n}. Since $\text{pref}(s,t) \leq |s|$, we can conclude that $\text{opt}(TSP) \leq \text{opt}(S)$, where $\text{opt}(TSP)$ 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 have to 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. A minimum-cost cycle cover can be computed exactly in $O(n^3)$ time by a well-known reduction to the assignment problem [9]. Since a tour is a cycle cover, $\text{opt}(C) \leq \text{opt}(TSP)$. 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 could also weight the edges by their overlap, find a maximum-cost cycle cover and get 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. [14, 13, 7]) than approximating the shortest superstring. We now describe a generic superstring algorithm used, in some form, by [1],[13] and [2]. An execution of the algorithm appears as Figure 1.

GENERIC SUPERSTRING ALGORITHM
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 [13].

We now analyze the generic algorithm in a way that anticipates our improvements. A more detailed analysis appears in [1]. Let $d(C)$ be the sum of the distances and $\text{ov}(C)$ be the sum of the overlaps of the edges in a cycle cover. Consider the second cycle cover CC. Let $\text{opt}(R)$ be the optimal superstring on the strings in $r_c \in R$ and observe that $\text{opt}(R) \leq \text{opt}(S)$. Let $\bar{\alpha}$ be the string produced in Step 5, a superstring of R. Since the shortest superstring for R is a cycle cover for G', the cycle cover finds as least as much overlap as the optimal amount of overlap for R, which we will denote $\text{opt}(\text{ov}(R))$. Thus $\text{opt}(\text{ov}(R)) \leq \text{ov}(CC)$. When we delete one edge from each cycle, we are giving up the overlap in that edge. Let ov^n_γ denote the overlap in the edge not taken, and let ov^t_γ denote the remaining overlap in γ. Let $\text{ov}^t = \sum_{\gamma \in CC} \text{ov}^t_\gamma$ and $\text{ov}^n = \sum_{\gamma \in CC} \text{ov}^n_\gamma$. Thus, $\bar{\alpha} \leq |R| - \text{ov}^t$. By definition, $|R| \leq \text{opt}(R) + \text{opt}(\text{ov}(R)) \leq \text{opt}(R) + \text{ov}(CC)$. Combining these two inequalities with $\text{ov}(CC) = \text{ov}^n + \text{ov}^t$, gives that $\bar{\alpha} \leq \text{opt}(R) + \text{ov}^n$. We then must extend each cycle, in Step 6. Let $\text{Ext}(\gamma)$ 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

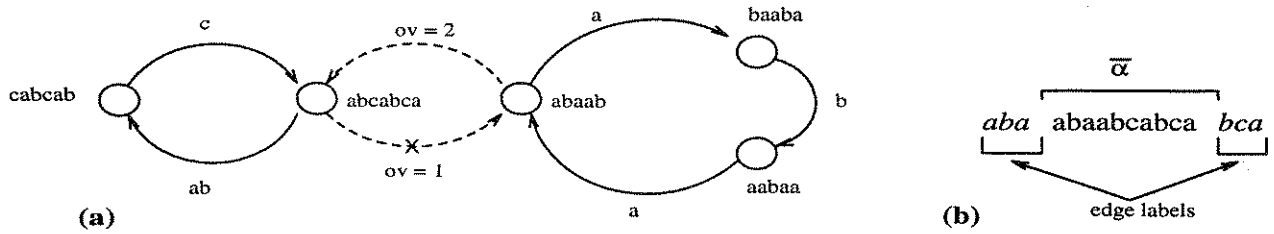

Figure 1: Execution of GENERIC SUPERSTRING ALGORITHM. Nodes are labeled with strings, edges with prefixes. (a) The graph after Step (4). Solid edges are in C, dashed edges in CC. The edge with an X is the one discarded in Step (4). (b) The final string consisting of $\bar{\alpha}$ (the merge of *abaab*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

string obtained, as

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

Let $d(c)$ be the sum of the weights of the edges of a cycle $c \in C$; so $d(C) = \sum_{c \in C} d(c)$. 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 $\text{ov}^n \leq d(C)$. Now, we extend $\bar{\alpha}$ by the edge labels on each cycle, adding a total of $d(C)$ to the length of the string. Let α be the resulting string. We conclude that

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

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

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 [13] in their $2\frac{8}{9}$ algorithm, if γ has three or more vertices, then $ov_\gamma^n \leq \frac{2}{3} \sum_{c \in \gamma} d(c)$.

Thus we only need to devise a better solution to the problem of 2-cycles in CC, if we are to get an approximation bound no less than $2\frac{2}{3}$. 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 $ov_\gamma^n + Ext(\gamma) \leq \beta(d(c_v) + d(c_t))$, for some $\beta \geq \frac{5}{3}$, is a $(1+\beta)$-approximation algorithm for the superstring problem.*

We will show in Section 4 that for each 2-cycle $\gamma \in CC$, either ov_γ^n or $\text{Ext}(\gamma)$ 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 $\text{gen}(\varsigma)$ be the string formed by an infinite repetition of ς. We say that s has *periodicity* x if there exists a string ς with $|\varsigma| = x$ such that s is substring of $\text{gen}(\varsigma)$. Let $\text{per}(c)$ 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 $\text{gen}(\text{per}(c))$. We define $\overset{\infty}{\text{per}}(c)$ to be $\text{gen}(\text{per}(c))$. Note that $\text{per}(c)$ must be irreducible; otherwise a cycle with less total distance could generate the same strings, contradicting the minimality of the cycle cover.

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

Corollary 2.3 *Let w be a substring of both $gen(\sigma_j)$ and $gen(\sigma_k)$. 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 if we want 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. In this section we present the 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. If either $\mathrm{ov}(v,t)$ or $\mathrm{ov}(t,v)$ is large but the other is small, then the obvious choice is the high-overlap edge. But if both edges have high overlap, we must discard one of them; these high-overlap 2-cycles are the bottleneck of the generic algorithm. We observe that both edges cannot participate in an optimal solution; in a sense this is evidence that the weight of the second cycle cover is a weak lower bound.

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

Consider the string z in Figure 2b. Here $\mathrm{per}(c) = ababadababadab$, $\sigma = ababad$, $y_\ell = abababadababadababa$ and $y_r = abababadababadabab$. So $|y_\ell|, |y_r| > \frac{3}{4}d(c) + \frac{3}{4}|\sigma|$, and we say that σ is a repeater of z. The idea is that if some other cycle c' has $\mathrm{per}(c') = \sigma$, then a string from c with σ as a repeater could form a high-overlap 2-cycle with the representative of c'. 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 you took σ and tried to repeat it as many times as possible, in both directions, while being consistent with c, you get y_σ. When the context is clear, we will drop the σ and just refer to witness y. In the example above $y = y_\ell$. 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. It can be verified that this simplification is conservative; using the maximal witness in our analysis does not yield a stronger result.

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.

Definition 3.3 Let c be a cycle with (g,h)-repeater σ and maximal witness y. Consider a copy of y in $\overset{\infty}{\mathrm{per}}(c)$. Call the point just to the left of the first character of y a *left discontinuity* and the point just to the right of the last character of y a *right discontinuity*. If $|y| < d(c)$ then the region between the right discontinuity and the left one is called a *positive anomaly* X_σ. If $|y| \geq d(c)$ then the two copies of y overlap and the region between the left and right discontinuities is called a *negative anomaly* X_σ.

We can picture the anomalies in a cycle c in terms of parentheses. Consider two copies of the maximal witness y in $\overset{\infty}{\mathrm{per}}(c)$, as in Figure 2b. We can mark the left and right ends of each copy of y with left and right parentheses. Then a negative anomaly is enclosed by a matching pair of parentheses. In Figure 2b, the negative anomaly is (aba). Intuitively, if a string in this cycle contains the anomaly X_σ near either end, then

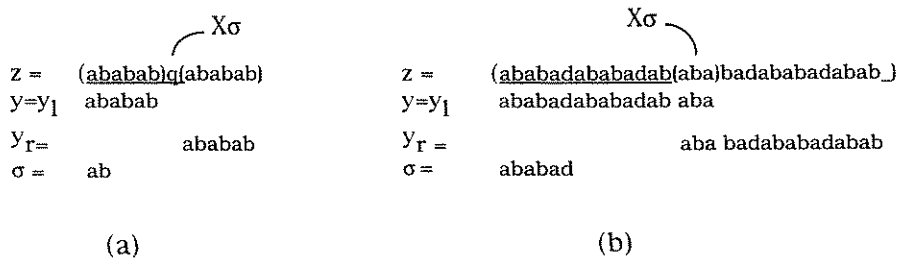

Figure 2: Positive and Negative Anomalies. Per(c) is underlined. Left discontinuities are indicated by a '(', and right discontinuities by a ')'. (a) shows a positive anomaly. (b) shows a negative anomaly. y and σ are also shown.

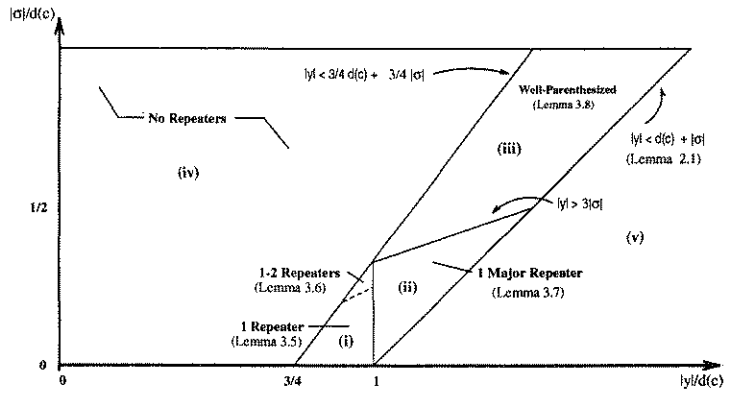

Figure 3: Map of 2-Cycle Structure. All repeaters are $(\frac{3}{4}, \frac{3}{4})$-repeaters.

it cannot attain high overlap with another representative whose period is σ. A positive anomaly 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 case q is the positive anomaly. Each anomaly appears once every $d(c)$.

We can identify the possible relationships which the various anomalies may have with respect to each other. Let $\sigma_1, \sigma_2, \ldots, \sigma_m$ be the repeaters in some cycle. We say that two anomalies X_{σ_i}, X_{σ_j} are *properly nested* if X_{σ_i} is a negative anomaly and X_{σ_j} falls strictly within the region of X_{σ_i}. We say that two anomalies X_{σ_i}, X_{σ_j} are *disjoint* if both endpoints of X_{σ_i} are to the right of both endpoints of X_{σ_j}, or visa versa.

We will show that any two anomalies of $(\frac{3}{4}, \frac{3}{4})$-repeaters are either nested or disjoint. This implies that any set of anomalies is well parenthesized. (Here we take the view described above that negative anomalies are pairs of parentheses and positive anomalies are single entities.) We will call X_σ a *major anomaly* if it is not nested within any other anomaly. A *minor anomaly* X_σ is one which is nested within some negative anomaly. We will sometimes refer to a *major* or *minor repeater* σ, meaning that X_σ is a major or minor anomaly.

This parenthesis structure naturally gives rise to a forest representation of the nesting of minor anomalies within major ones. We can view each major anomaly as the root of a parse tree, and each minor anomaly as another node in the tree according to its nested position, as pictured in Figure 7. Parent-child relationships are thus well-defined and we have a tree for each major anomaly. We will use this representation in our algorithm to handle minor anomalies.

The properties of high-overlap two-cycles in the second cycle cover CC, taken together, form a map from which we can read the possible structure of a two-cycle for a given set of parameters. In certain regions, the picture is more structured as the number of anomalies may be limited or the structure may be more clearly identified. We display this map in Figure 3, which is a two-dimensional plot of what is actually a three-dimensional space. We have labeled the axes by $|y|$ and $|\sigma|$, normalized by $d(c)$. Consider a cycle c and let σ be the largest (in terms of $|\sigma|$) $(\frac{3}{4}, \frac{3}{4})$-repeater of c_t, and let y be its witness. In order to characterize c on this graph, plot the point $(|y|/d(c), |\sigma|/d(c))$.

The rest of the section will be devoted to describing how the our choice of representative is affected by

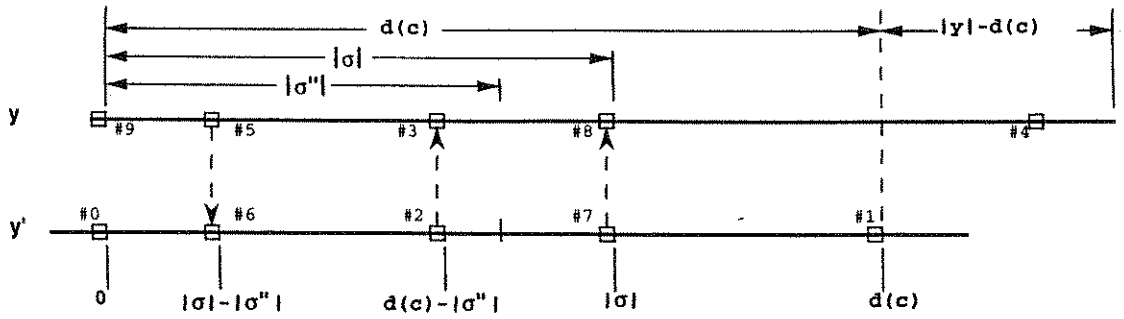

Figure 4: Proof of Lemma 3.7, part 3

the region in which a cycle falls. If a cycle falls in Region (iv) its representative cannot participate in a high-overlap 2-cycle. In Region (i) which has relatively small σ and/or long witness y, we are able to prove strong uniqueness properties. In Region (ii) only a simple parenthetical structure is possible. In Region (iii) a more complicated parenthetical structure can arise, and we are able to exploit this structure to extend 2-cycles inexpensively.

We can now state some properties of (g,h)-repeaters and their anomalies. Representative proofs appear in the Appendix. We first show that for values of g and h which include the lower portion of both Region (i) and Region (ii), there can be only one (g,h)-repeater. Our analysis will use this uniqueness to bootstrap to larger values of $|\sigma|$.

Lemma 3.4 *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.*

For a range of parameters which includes the rest of Region (i), we can show that a cycle can have no more than two repeaters. We can also show that in Region (ii) there can only be one major anomaly, although nested minor anomalies are possible.

Lemma 3.5 *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 $|\sigma_i| \leq \frac{1}{3}d(c), i \in \{1,2\}$.*

Lemma 3.6 *Let σ be the largest (in terms of $|\sigma|$) $(\frac{3}{4},\frac{3}{4})$-repeater in cycle c, and its witness y with $|y| \geq d(c)$ and $|y| \geq 3|\sigma|$. Then if there is some other repeater σ' in c, $X_{\sigma'}$ is strictly contained within X_σ.*

In Region (iii), where the size of $|\sigma|$ is large relative to $d(c)$ and $|y|$, we have no useful bound on the number of major $(\frac{3}{4},\frac{3}{4})$-repeaters. However, as discussed above we can show that major and minor anomalies in this region are well parenthesized and thus are naturally represented as a forest. The proof of Part 1 is a simple application of Corollary 2.3. We include below the proof of Part 3, which illustrates an important technique; the proof of Case 2 is similar.

Lemma 3.7 *Let σ,σ' be $(\frac{3}{4},\frac{3}{4})$-repeaters in cycle c, $|\sigma| > |\sigma'|$, with witnesses y, y'. Then*

1. *If both $y, y' \leq d(c)$, then X_σ and $X_{\sigma'}$ are disjoint.*
2. *If $y > d(c)$, $y' \leq d(c)$, then either X_σ and $X_{\sigma'}$ are disjoint or $X_{\sigma'}$ is properly nested within X_σ*
3. *If both $y, y' > d(c)$, then X_σ and $X_{\sigma'}$ are disjoint or properly nested.*

Proof: Part 3. Suppose for purpose of contradiction that the two negative anomalies X_σ and $X_{\sigma'}$ are neither disjoint nor properly nested. Then Figure 4 reflects the situation; in particular note that a copy of y and of y' must overlap by at least $d(c)$ in $\overset{\infty}{per}(c)$. We will consider the region of size $d(c)$ which starts with the first character of y. Our strategy will be to show that the first character to the left of the overlap region (labeled "position # 0" in the Figure) is the next character which is required to extend y one more character to the left, contradicting y being the maximal witness for σ.

We will accomplish this by a series of shifts. When we shift by $|\sigma|$ or $|\sigma'|$ from some position j to position $j+1$ both positions must be contained within the same copy of y and y' respectively. We may also be allowed to shift by multiples of $|\sigma'|$ as long as we stay within y' when doing so; this larger period of y' we will refer to as σ''; we may assume that $|\sigma''| \geq \frac{1}{2}d(c)$. For ease of exposition we will use the convention that we will shift by $|\sigma|$ in the string marked y and by $|\sigma''|$ in the string marked y', and when necessary we will "cross" to the appropriate string. Of course we can shift by $d(c)$ anywhere in $\overset{\infty}{\text{per}}(c)$.

Following is the sequence of positions, with reference to Figure 4; "+" moves are to the right, "−" moves are to the left. Starting at Position #0 in y', we move $+d(c)$ to #1, $-|\sigma''|$ to #2, cross to #3, $+|\sigma|$ to #4, $-d(c)$ to #5, cross to #6, $+|\sigma''|$ to #7, cross to #8, and $-|\sigma|$ to #9.

It remains to show that these moves are legal; that is, that each shift remains within the range of the appropriate witness. While Figure 4 is only representative, that is different values of $|\sigma|$ and $|\sigma'|$ yield slightly different pictures, most of the positions are easily verified to be within bounds. The one which might be an issue is from Position #3 to #4. Here it is necessary that $|y| - d(c) \geq |\sigma| - |\sigma''|$. Supposing the contrary, we get $|\sigma| - |\sigma''| > y - d(c) > \frac{3}{4}d(c) + \frac{3}{4}|\sigma| - d(c)$. Solving for $|\sigma''|$ we get $|\sigma''| < \frac{1}{2}d(c)$ but we know $|\sigma''| \geq \frac{1}{2}d(c)$. ■

4 The Algorithm

Our algorithm 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 anomalies.

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 $\text{Ext}(v, w) = |\beta| - |\langle v, w\rangle|$. If γ is the two-cycle containing v and w, we will sometimes also write also write $\text{Ext}(\gamma)$, meaning $\min(\text{Ext}(v, w), \text{Ext}(w, v))$. (This is the notation we introduced informally in Section 2. By the discussion in Section 2 we know that $\text{Ext}(v, w) \leq d(v) + d(w)$). 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 $\text{Ext}(\gamma)$ 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:
  (a) if γ is a 2-cycle (v,t) such that $\min(\text{ov}(v,t), \text{ov}(t,v)) > \frac{3}{4}(d(c_v) + d(c_t))$
    if $\text{Ext}(v,t) \leq \text{Ext}(t,v)$
      then Discard edge (t,v) ; Extend $\langle v, t\rangle$ by $\text{Ext}(v,t)$
      else Discard edge (v,t); Extend $\langle t, v\rangle$ by $\text{Ext}(t,v)$
  (b) Otherwise discard edge of cycle γ with least overlap; Extend each vertex w by $d(c_w)$
(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 anomalies. Our first choice for representative is a string which falls in Region (iv) and hence does not contain a $(\frac{3}{4}, \frac{3}{4})$-repeater; such a string cannot achieve high overlap on both edges of a two-cycle. It may be that every string in the cycle has a $(\frac{3}{4}, \frac{3}{4})$-repeater; in this case we use a more precise notion of the relationship between a string and the anomalies in its cycle.

Definition 4.1 Suppose a cycle c contains an anomaly X_σ. Consider a string z in c. If X_σ is positive, we say that z *touches* X_σ each time it passes through X_σ or includes at least one character of X_σ. If X_σ is negative, we say that z *touches* X_σ each time that X_σ is strictly included in z.

For each major anomaly X_{σ_i}, let t_i be the maximum number of times any string touches σ_i. If a string touches each anomaly exactly t_i times, then it is called a *consensus candidate*. If every string in the cycle has a $(\frac{3}{4}, \frac{3}{4})$-repeater, then our next choice is a consensus candidate. The choice of a consensus candidate guarantees that the representative covers most of the total length required by all of the strings in the cycle, and our cost of extension will be low.

The cycle may not contain a consensus candidate. It may be that one string s will touch one anomaly X_σ the most times, while another string t touches X_σ one less time but touches $X_{\sigma'}$ one more time than s. In this case we choose a string which has the largest $(\frac{3}{4}, \frac{3}{4})$-repeater.

In either of the latter two cases, we might have more than one string which meet our criteria in terms of major anomalies. We break ties in this case by examining the nested minor anomalies in a similar manner. Given the parse tree representation of this nesting described in Section 3, the method for doing so can be easily described. If we find no way to break ties even with the minor anomalies, 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 anomaly X_σ, we will use the notation $\sigma(w)$ to refer to σ.

```
PROCEDURE FINDREPS (c)
Find all (3/4, 3/4)-repeaters and associated anomalies in c
Label major anomalies X_{σ_1}, X_{σ_2}, ..., X_{σ_r} such that |σ_1| > |σ_2| > ... > |σ_r|
Build the parse tree for each major anomaly X_{σ_i}
if there are one or more strings without a (3/4, 3/4)-repeater
    then r_c ← longest such string
    else if there is at least one consensus candidate
            then let X be the major anomaly in which all consensus candidates begin
            else let X be X_{σ_1}
        Let a be the leftmost string in X; let z be the rightmost string in X
        if a ≠ z
            then r_c ← TIEBREAK(a, z, X)
            else r_c ← a
```

```
PROCEDURE TIEBREAK(A,Z,X)          {X is a parse tree}
w ← 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 |σ(anc_a)| ≥ |σ(anc_z)|
            then Return a
            else Return z
```

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, $\text{ov}^n_\gamma + \text{Ext}(\gamma) \leq \beta(d(c_v) + d(c_t))$.

Lemma 4.2 *Let γ be a 2-cycle in CC with v the representative of cycle c_v and t the representative of c_t. Then*

$$ov^n_\gamma + Ext(\gamma) \leq \frac{7}{4}(d(c_v) + d(c_t)).$$

Proof: Assume without loss of generality $d(c_t) \geq d(c_v)$. If $\min(\text{ov}(v,t), \text{ov}(t,v)) \leq \frac{3}{4}(d(c_v) + d(c_t))$ then $\text{ov}^n_\gamma \leq \frac{3}{4}(d(c_v) + d(c_t))$ and we can achieve the stated bound with an extension cost of $d(c_v) + d(c_t)$. 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 $\text{per}(v) = \sigma$ for any $(\frac{3}{4}, \frac{3}{4})$-repeater in t.

Now suppose $\min(\text{ov}(v,t), \text{ov}(t,v)) > \frac{3}{4}(d(c_v) + d(c_t))$. Since v achieves this much overlap at both ends of t, there must be at least one $(\frac{3}{4}, \frac{3}{4})$-repeater σ' in c_t, and in fact $\sigma' = per(v)$. 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.

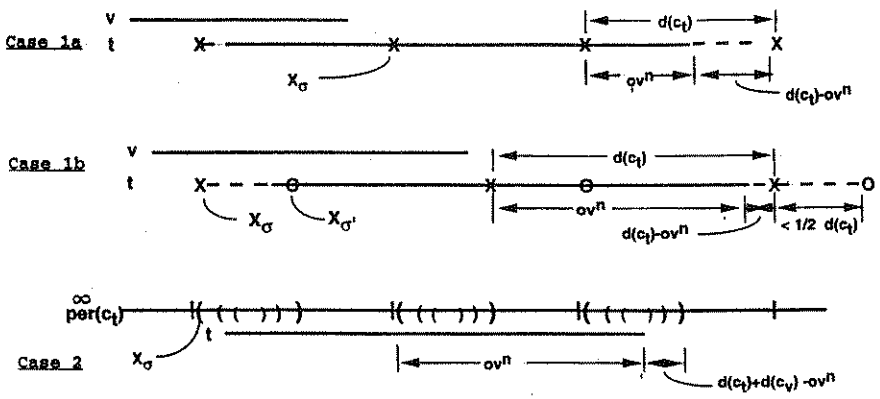

Figure 5: Cases 1a, 1b, and 2 from Proof of Lemma 4.2.

Let σ be the largest $(\frac{3}{4}, \frac{3}{4})$-repeater found in FINDREPS(c_t). We will consider three cases as described in Figure 3, determined by the size of the largest $(\frac{3}{4}, \frac{3}{4})$-repeater σ found in c_t and by the length of its maximal witness y:

1. $\frac{3}{4}(d(c) + |\sigma|) \leq |y| \leq d(c)$
2. $|y| \geq d(c)$ and $|y| \geq 3|\sigma|$
3. $\frac{3}{4}(d(c) + |\sigma|) \leq |y| \leq 3|\sigma|$

Case 1. Since $|y| \leq d(c)$, $|\sigma| \leq \frac{1}{3}d(c)$ and by Lemma 3.5 there can be at at most two $(\frac{3}{4}, \frac{3}{4})$-repeaters in c. First suppose there is only one, say σ. Then FindReps(c) must have chosen as t a string that had more touches of X_σ than any other string in c. Since t could only contain σ as a $(\frac{3}{4}, \frac{3}{4})$-repeater, $\sigma = \text{per}(v)$.

To extend t to include another string t' in c_t, we align the copies of X_σ in t' with those in t. By our choice of t, t' can't extend beyond the next copy of X_σ (the rightmost line in Figure 5 Case 1a). By the definition of anomaly, the overlap with v cannot touch X_σ. So the remaining cost of extending to include t' is at most $d(c_t) - \text{ov}^n_\gamma$. We still have to extend to include other strings in c_v; that requires no more than $d(c_v)$, giving us

$$\text{ov}^n_\gamma + \text{Ext}(\gamma) \leq \text{ov}^n_\gamma + d(c_t) - \text{ov}^n_\gamma + d(c_v) = d(c_t) + d(c_v).$$

Suppose we had two $(\frac{3}{4}, \frac{3}{4})$-repeaters in c_t with anomalies X_σ and $X_{\sigma'}$. If in FINDREPS(c_t) we chose t because it had the most touches of both anomalies, the analysis would be the same as that above with only one $(\frac{3}{4}, \frac{3}{4})$-repeater. Suppose there was not such a string, 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. (See Figure 5.) The extension cost to the next copy of X_σ is $d(c_t) - \text{ov}^n_\gamma$. However, since some t' might include one more copy of X_σ, we also might have to extend to the next copy of $X_{\sigma'}$. Because ALGORITHM SHORTSTRING extends on the less expensive side, this distance is no more than $\frac{1}{2}d(c_t)$. Again assuming that v will have to be extended by its full period, we have

$$\text{ov}^n_\gamma + \text{Ext}(\gamma) \leq \text{ov}^n_\gamma + d(c_t) - \text{ov}^n_\gamma + \frac{1}{2}d(c_t) + d(c_v) = d(c_t) + d(c_v) + \frac{1}{2}d(c_t) < \frac{17}{12}(d(c_t) + d(c_v)).$$

This last bound follows since by Lemma 3.4 there can only be one $(\frac{1}{2}, 2)$-repeater in c_t. So for the larger of the $(\frac{3}{4}, \frac{3}{4})$-repeaters, it must be that $\frac{3}{4}(d(c_t) + d(c_v)) < |y| < \frac{1}{2}(d(c_t) + 2d(c_v))$ which solves to $d(c_v) > \frac{1}{5}d(c_t)$, so $d(c_t) < \frac{5}{6}(d(c_t) + d(c_v))$.

Case 2. In this case we have $|y| \geq d(c)$ and $|y| \geq 3|\sigma|$ which are the preconditions required for Lemma 3.6. So we know that X_σ is the only major anomaly, and it might be positive or negative. If it is negative, it might have minor anomalies nested within in. If so, however, by Lemma 4.4 they must be simply nested, not side by side. (See Figure 5.) Since there was not a string in c_t without a $(\frac{3}{4}, \frac{3}{4})$-repeater, there must be at least two strings t and t' which start and end in X_σ, and so TIEBREAK(t, t', X_σ) must have been called.

In general 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*. Since in Region (ii) minor

anomalies can only be simply nested, TieBreak must have broken the tie between t and t' strongly. Therefore regardless of whether $\mathrm{per}(v)$ is actually σ or some σ' whose anomaly $X_{\sigma'}$ is nested within X_σ, the overlap begins within one copy of X_σ and cannot extend past the next copy. Also note that no t'' can extend past $X_{\sigma'}$, or it would have had more touches of X_σ than t. Our analysis then is

$$\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) \leq \mathrm{ov}_\gamma^n + d(c_t) + d(c_v) - \mathrm{ov}_\gamma^n + d(c_v) = d(c_t) + 2d(c_v) \leq \frac{4}{3}(d(c_t) + d(c_v)).$$

This last follows because in this case $d(c_v) \leq \frac{1}{2}d(c_t) \leq \frac{1}{3}(d(c_t) + d(c_v))$.

Before proceeding to Case 3 we need some further details about the parenthesization of anomalies in the remaining region, where $|\sigma|$ is larger. The proofs of these Lemmas follow in a straightforward way from the definition of $(\frac{3}{4}, \frac{3}{4})$-repeaters and anomalies, and the application of Corollary 2.3.

Lemma 4.3 *Let X_σ be a major anomaly in cycle c and $X_{\sigma'}$ a minor anomaly nested within X_σ. Then $|\sigma| \geq \frac{3}{5}d(c)$.*

Lemma 4.4 *If two minor anomalies $X_{\sigma'}$ and $X_{\sigma''}$ are nested separately within another anomaly X_σ, then $|\sigma| \geq \frac{3}{5}d(c)$, and $|\sigma'| + |\sigma''| \geq d(c)$.*

Lemma 4.5 *Let X_σ be a major anomaly in cycle c and $X_{\sigma'}$ a minor anomaly nested within X_σ. Then if a string t is a witness to both σ and σ' then $d(y, y') \geq \frac{1}{2}d(c) - |\sigma'|$.*

Lemma 4.6 *Let X_σ be any negative anomaly in cycle c. Then $|X_\sigma| < |\sigma|$.*

Case 3. Recall that in this case we have $\frac{3}{4}(d(c) + |\sigma|) \leq |y| \leq 3|\sigma|$. In this region we may have several major anomalies as well as minor anomalies nested according to the constraints of Lemmas 3.7 and 4.4.

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. It also could have had ties between itself and another string t' broken strongly or weakly. We consider each of the resulting four cases; the first appears below and the remainder in the Appendix.

Subcase A. Suppose t was a consensus candidate, and any ties were broken strongly. Let $X_{\sigma'}$ be the major anomaly in which t begins and ends. (See Figure 6.) If $\mathrm{per}(v) = \sigma'$, and using Lemma 4.6, then

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

It could be that $\mathrm{per}(v) = \sigma''$, with $X_{\sigma''}$ nested within $X_{\sigma'}$. In this case we apply Lemma 4.5 and get

$$\begin{aligned} \mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) &\leq \mathrm{ov}_\gamma^n + 2d(c_t) - \mathrm{ov}_\gamma^n - (\frac{1}{2}d(c_t) - |\sigma'|) + d(c_v) \\ &\leq 2(d(c_t) + d(c_v)) - \frac{1}{2}d(c_t) \\ &\leq \frac{7}{4}(d(c_t) + d(c_v)). \end{aligned}$$

We now combine Lemmas 2.2 and Lemma 4.2 with the results of Section 3 to obtain:

Theorem 4.7 *Algorithm ShortString is a $2\frac{3}{4}$-approximation for the shortest superstring problem.*

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 [7], 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, 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 First 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] R. Kosaraju, J. Park, and C. Stein. Long tours and short superstrings. To appear in FOCS 94, May 1994.

[8] 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.

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

[10] 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.

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

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

[13] 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.

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

Appendix

Representative Proofs of Results of Section 3

Following are proofs of some of the technical results of Section 3.

Lemma 3.4 *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

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

These lower bounds on length guarantee that

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

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

Lemma 3.5 *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 Lemma 2.3.

Consider the three overlaps in the order $\text{ov}(y_1, y_2)$, $\text{ov}(y_2, y_3)$, $\text{ov}(y_3, y_1)$. We have $\text{ov}(y_i, y_j) = |y_i| - d(y_i, y_j)$. Summing the three overlaps and applying the definition of $(\frac{3}{4}, \frac{3}{4})$-repeater gives us

$$\begin{aligned} \sum_{i,j} \text{ov}(y_i, y_j) &= \sum_i |y_i| - d(c) \\ &\geq \frac{9}{4}d(c) + \frac{3}{4}\sum_i |\sigma_i| - d(c) \\ &= \frac{5}{4}d(c) + \frac{3}{4}\sum_i |\sigma_i| \end{aligned}$$

To avoid a contradiction by Corollary 2.3, we would need for each pair (σ_i, σ_j) to sum to more than the overlap of their witnesses. Summing over the three overlaps we have

$$2\sum_i |\sigma_i| > \frac{5}{4}d(c) + \frac{3}{4}\sum_i |\sigma_i|$$

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

Remaining Cases in Proof of Lemma 4.2

In Section 4 we proved the first of four subcases of Case 3 of the analysis. Recall that in this case we know $\frac{3}{4}(d(c)+|\sigma|) \leq |y| \leq 3|\sigma|$. Our four subcases we determined by whether t was chosen as a consensus candidate or because it had the largest $(\frac{3}{4}, \frac{3}{4})$-repeater, and by whether any ties were broken strongly or weakly.

Subcase B. 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 anomaly before the next copy of X_σ or such a string would have had the maximum number of touches of each major anomaly and would have been made the representative. (See Figure 6.) First suppose that $\mathrm{per}(v) = \sigma$. Because we're in case 3, we know that $|\sigma| \geq \frac{1}{3}d(c_t)$ and

$$\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) \leq \mathrm{ov}_\gamma^n + 2d(c_t) - \mathrm{ov}_\gamma^n + d(c_v) \leq 2d(c_t) + d(c_v) \leq \frac{7}{4}(d(c_t) + d(c_v))$$

It could also be that $\mathrm{per}(v) = \sigma'$, with $X_{\sigma'}$ nested within X_σ. The analysis in this case follows exactly that of Case A, giving us a bound of $\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) \leq \frac{7}{4}(d(c_t) + d(c_v))$

Subcase C. Now suppose t was a consensus candidate, and there was a tie which was broken weakly. Suppose t started and ended in major anomaly $X_{\sigma'}$. Then any other string t' in the cycle can be aligned with t and not extend beyond the end of $X_{\sigma'}$. (See Figure 6.) By Lemma 4.4 we know that $|\sigma| \geq \frac{3}{5}d(c_t)$, and $|\sigma'| \geq \frac{1}{2}d(c_t)$. We might have $\mathrm{per}(v) = \sigma$ or $\mathrm{per}(v) = \sigma'$; in either case we have

$$\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) \leq \mathrm{ov}_\gamma^n + d(c_t) + |\sigma| - \mathrm{ov}_\gamma^n + d(c_v) \leq 2d(c_t) + d(c_v) \leq \frac{5}{3}(d(c_t) + d(c_v))$$

Subcase D. 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 Case B, no other string t' can touch the last major anomaly before the next copy of X_σ, or it would have been chosen representative on the basis of maximum touches. Since we know that $d(c_v) \geq \frac{1}{2}d(c_t)$,

$$\mathrm{ov}_\gamma^n + \mathrm{Ext}(\gamma) \leq \mathrm{ov}_\gamma^n + 2d(c_t) - \mathrm{ov}_\gamma^n + d(c_v) \leq \frac{5}{3}(d(c_t) + d(c_v)$$

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

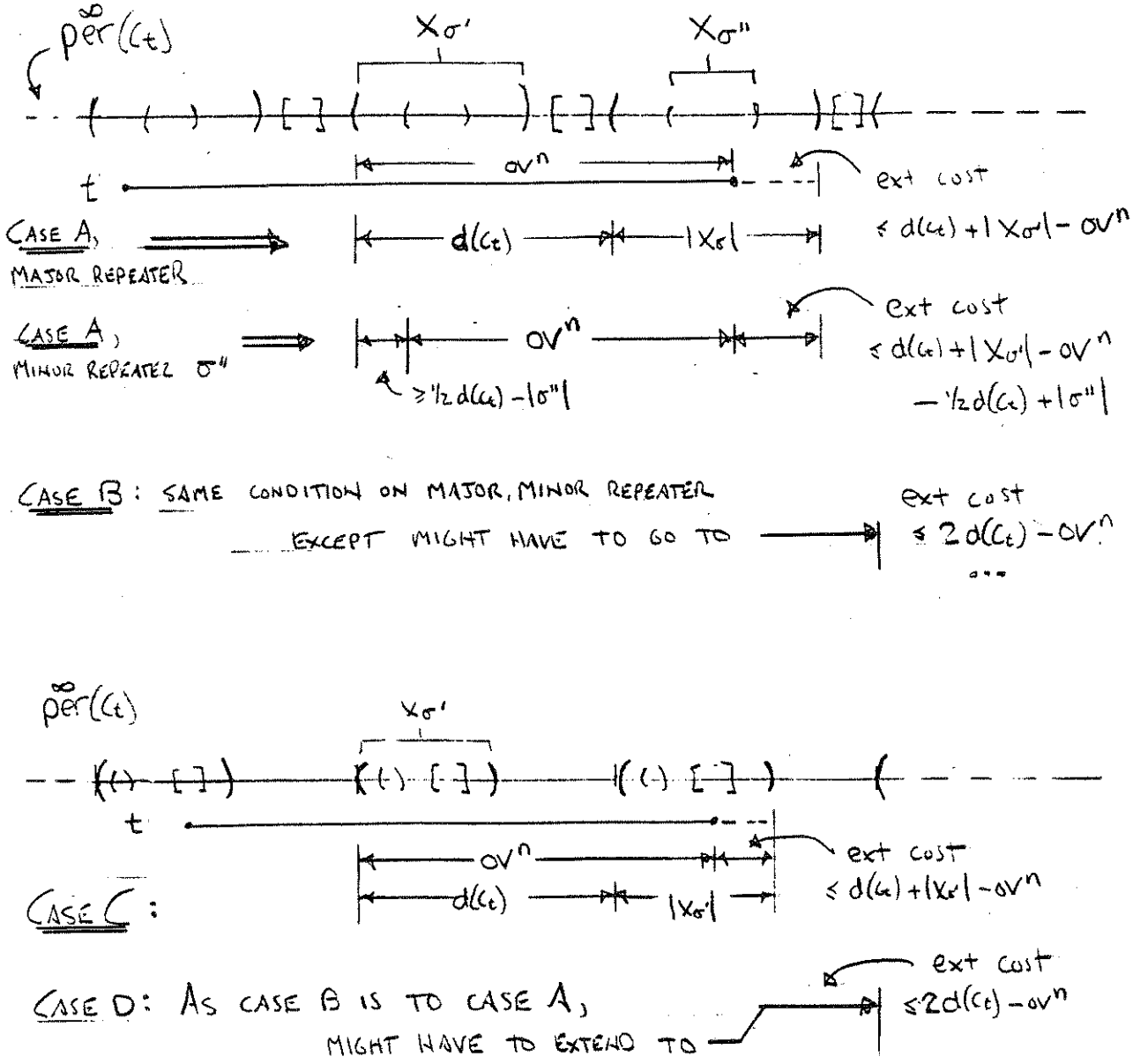

Figure 6: Proof of Lemma 4.2 Case 3A-D.

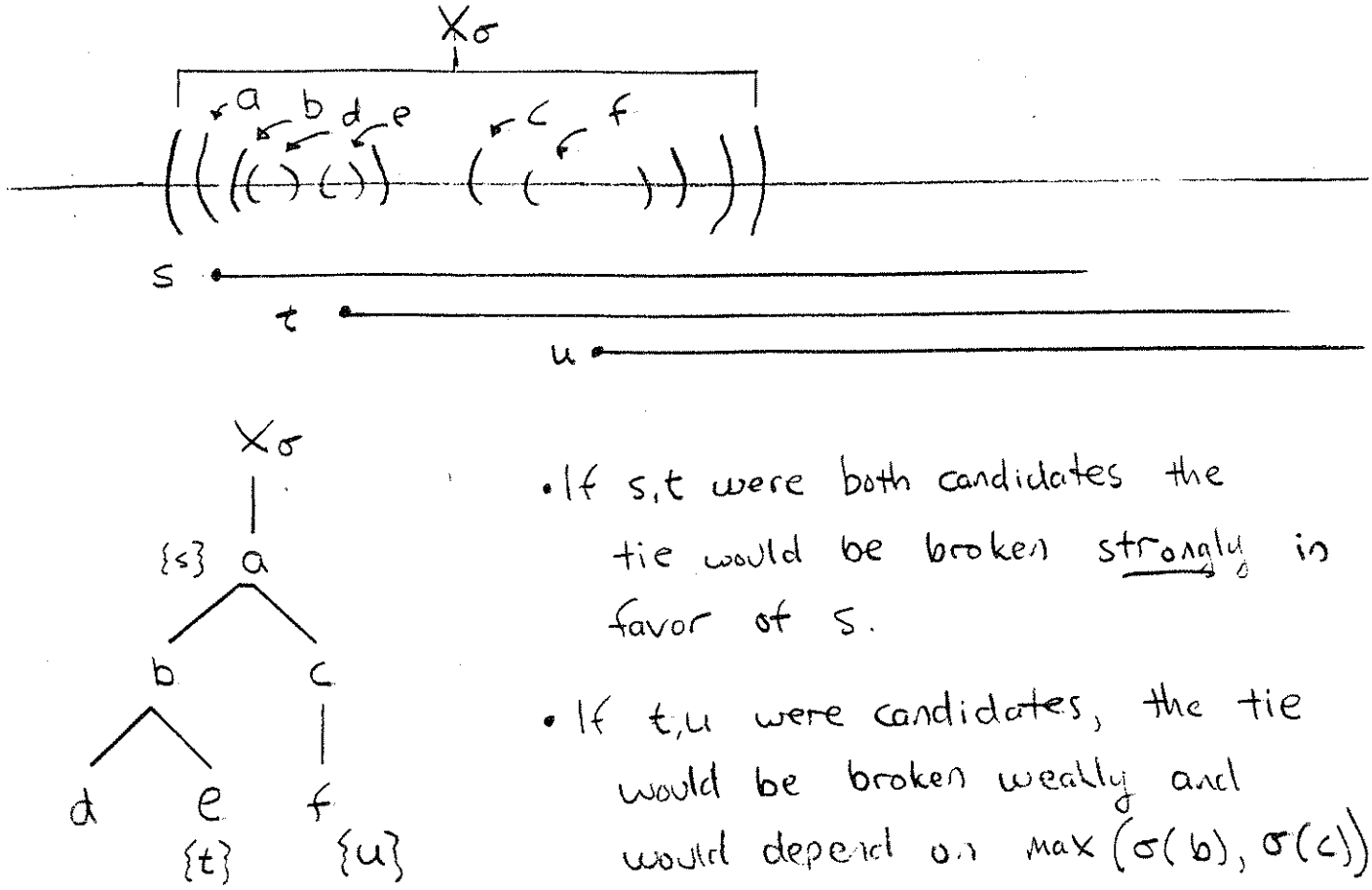

Figure 7: Parse Tree Representation of Anomalies