Rotations of Periodic Strings and Short Superstrings

Tao Jiang
University of California - Riverside

Joint work with

Dany Breslauer
MPI für Informatik

Zhigen Jiang
McMaster University

The Shortest Superstring Problem (I)

Given a set of strings $S = \{s_1, \ldots, s_m\}$, find a shortest superstring $s$ that contains all $s_i$ as substrings.

Example. "Alf ate half lethal alpha alfalfa."

$S = \{\text{alf, ate, half, lethal, alpha, alfalfa}\}$

$w_1 = \text{atehalflethalalphaalfalfa}$ \hspace{1em} |w_1| = 25

$w_2 = \text{lethalalphaalfate}$ \hspace{1em} |w_2| = 17

$w_3 = \text{lethalalphaalfate}$ \hspace{1em} |w_3| = 17

Assume the input set $S$ is substring-free.

The Shortest Superstring Problem (II)

Applications:

1. Data compression. A set of strings can be represented by a superstring and positions of the strings in the superstring and their lengths.

2. DNA sequencing. State-of-the-art biochemistry can sequence a fragment of about 500 nucleotides. Longer DNA molecules are "cut" into short overlapping fragments that are sequenced separately. These fragments are then "assembled" by a shortest superstring algorithm.

The Shortest Superstring Problem (III)

The shortest superstring problem is MAX-SNP hard [GMS80,B].

Try to approximate!

Denote as $\text{opt}(S)$ the length of a shortest superstring and as $\text{maxov}(S)$,

$$\text{maxov}(S) = \sum_{s \in S} |s| - \text{opt}(S),$$

the compression or total overlap between strings in a shortest superstring.

$\text{maxov}(S)$ is the maximum overlap in any superstring!

Overlap approximation seems to be easier than length approximation.

[TU88] and [T89] prove that "the GREEDY" algorithm $\frac{1}{2}$-approximates the overlap.

Conjecture: "The GREEDY" algorithm 2-approximates the length. [BJLTY94] prove that it 4-approximates the length.
The Shortest Superstring Problem (IV)

<table>
<thead>
<tr>
<th>Length Approximation Algorithms</th>
<th>91</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blum, Jiang, Li, Tromp and Yannakakis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Teng and Yao</td>
<td>93</td>
<td>2.89</td>
</tr>
<tr>
<td>Czumaj, Gasieniec, Piotrow and Rytter</td>
<td>94</td>
<td>2.83</td>
</tr>
<tr>
<td>Kosaraju, Park and Stein</td>
<td>94</td>
<td>2.79</td>
</tr>
<tr>
<td>Armen and Stein</td>
<td>95</td>
<td>2.75</td>
</tr>
<tr>
<td>Armen and Stein</td>
<td>96</td>
<td>2.67</td>
</tr>
<tr>
<td>This work</td>
<td>96</td>
<td>2.67</td>
</tr>
<tr>
<td>This work</td>
<td>96</td>
<td>2.596</td>
</tr>
<tr>
<td>Sweedyk</td>
<td>96</td>
<td>2.5 ????</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Overlap Approximation Algorithm</th>
<th>94</th>
<th>0.603</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kosaraju, Park and Stein</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Basic Notations and Facts

Given strings \( s \) and \( t \), let \( y \) be the longest string such that \( s = ay \) and \( t = yz \), for some non-empty \( s \) and \( z \).

\[
s = \text{lethal} \quad \text{half} = t
\]

\[
\alpha(v(s, t)) = 3
\]

\[
\rho(s, t) = \text{let}
\]

\[
d(s, t) = 3
\]

\[
\langle s, t \rangle = \text{lethalhalf}
\]

\[
\langle s_i, \ldots, s_i \rangle \) is the shortest string containing the strings in the specified order:
\[
\langle s_i, \ldots, s_i \rangle = \rho(s_1, s_i) \cdots \rho(s_{i-1}, s_i) s_i
\]

\[
\langle \text{lethal, half, alfalfa} \rangle = \text{lethalalfalfa}
\]

Claim: The shortest superstring for \( S = \{s_1, \ldots, s_m\} \) is \( \langle s_{\pi(1)}, \ldots, s_{\pi(m)} \rangle \) for some permutation \( \pi \).

\[
\max_{\pi} = \sum_{i=1}^{m} |s_i| - \text{opt}(S) = \sum_{i=1}^{m-1} |\alpha(s_{\pi(i)}, s_{\pi(i+1)})|
\]

The Distance Graph

Given a set of strings \( S = \{s_1, \ldots, s_m\} \), the distance graph \( G_S \) has \( m \) vertices \( s_1, \ldots, s_m \).

The edge \((s_i, s_j), i \neq j\), has weight \( d(s_i, s_j) \).

Example: \( G_S \) for \( S = \{\text{ate, lethal, alpha, alfalfa}\} \):

![Distance Graph Example](image)

TSP\((G_S)\) is the minimum weight of any Hamiltonian cycle in \( G_S \) (i.e. optimal TSP solution).

Then, for each \( s_i \in S \),

\[
\text{TSP}(G_S) \leq \text{opt}(S) \leq \text{TSP}(G_S) + |s_i|
\]

\[
\text{opt}(S) \Leftrightarrow \rho(s_1, s_{\pi(2)}) \cdots \rho(s_{\pi(k-1)}, s_{\pi(k)}) s_{\pi(m)}
\]

\[
\text{TSP}(S) \Leftrightarrow \rho(s_1, s_{\pi(2)}) \cdots \rho(s_{\pi(k-1)}, s_{\pi(k)})(\rho(s_{\pi(m)}, s_{\pi(1)})
\]

Cycle Covers

A cycle cover of a graph is a collection of disjoint cycles that cover all vertices.

\( \text{CYC}(G_S) \) is the minimum weight of any cycle cover of \( G_S \).

Clearly \( \text{CYC}(G_S) \leq \text{TSP}(G_S) \leq \text{opt}(S) \).

\[
\text{opt}(S) \leq ?\text{CYC}(G_S)\).
\]

Good news:

We can compute \( \text{CYC}(G_S) \) in polynomial time.

This will be taken for granted in this talk.
Periods and Rotations of Strings

A string $s$ has a factor $x$ if $s = xy$, for some integer $i$ and prefix $y$ of $x$.

The factor of $s$, $f(s)$, is the shortest factor of $s$.

Denote the period of $s$ as $p(s) = |f(s)|$.

Two strings $s$ and $t$ are equivalent if $f(s)$ is a rotation of $f(t)$. Namely if $f(s) = xy$ and $f(t) = yx$.

The factor $x = f(s)$ of a semi-infinite string $s$ is the shortest string such that $s = x^\infty$.

Examples:
The string $aababa$ has factors $aab$, $aaba$, $abaaba$, and $aababa$.
The factor $f(aaba) = aab$.
The strings $aba$ and $baa$ are rotations of $aab$.
The string $(baa)^\infty$ is equivalent to $aababa$.

The Generic Superstring Algorithm (I)
First part:
1. Construct the distance graph $G_S$.
2. Find a minimum weight cycle cover $C$ in $G_S$.
3. Choose a representative string $t_c$ for each cycle $c \in C$, such that for some $j$:
   (a) $t_c$ contains $\langle s_{i_1}, \ldots, s_{i_j}, s_{i_1}, \ldots, s_{i_j} \rangle$, and
   (b) $t_c$ is contained in $\langle s_{i_1}, \ldots, s_{i_j}, s_{i_1}, \ldots, s_{i_j} \rangle$.
4. Let $T$ be the set of representatives above.

Remark: The strings in $T$ are pairwise inequivalent.

Lemma: $\text{opt}(T) \leq \text{opt}(S) + \text{CYC}(S) \leq 2\text{opt}(S)$.

Proof.
\[ \langle s_{i_1}, \ldots, s_{i_j}, s_{i_1}, \ldots, s_{i_j} \rangle = f(\langle s_{i_1}, \ldots, s_{i_j}, s_{i_1}, \ldots, s_{i_j} \rangle) s_{i_j} \]
\[ \text{opt}(\bigcup \{ s_{i_j} \}) \leq \text{opt}(S) \]

The Generic Superstring Algorithm (II)
Second part:
1. Construct the distance graph $G_T$.
2. Find a minimum weight cycle cover $C C$ in $G_T$.
3. Open each cycle of $C C$ arbitrarily.
4. Let $R$ be the set of the strings obtained above.
5. Concatenate the strings in $R$ to produce a superstring $s$ of $S$.

Overlap Lemma: [FW65] For inequivalent strings $s$ and $t$,
\[ \omega(s, t) \leq p(s) + p(t). \]

Let $OV$ be the overlap on the broken edges in all cycles of $C C$. Then,
\[ OV \leq \sum_{c \in C} p(t_c) = \sum_{c \in C} w(c) = \text{CYC}(G_S). \]

Conclusion: Recalling $\text{CYC}(G_T) \leq \text{opt}(T) \leq 2\text{opt}(S)$,
\[ |S| = \text{CYC}(G_T) + OV \leq 2\text{opt}(S) + \text{opt}(S) \leq 3\text{opt}(S). \]
The Overlap-Rotation Lemma

Let $\alpha = a_1 a_2 \cdots$ be a semi-infinite string.
Denote a rotation $\alpha[k] = a_k a_{k+1} \cdots$.
There exists an integer $k$, such that for any finite string $s$ that is inequivalent to $\alpha$ and satisfies $p(s) \leq p(\alpha)$,
$$ov(s, \alpha[k]) \leq \frac{2}{3}(p(s) + p(\alpha)).$$

Remarks:
The bound above is roughly tight as demonstrated by the string $\alpha = (010^{n+1})^\infty$.
$\alpha$ is semi-infinite for convenience.

The Improved Algorithm (I)

First part:
Choose the representatives $t_i$ with care.

Second part:
Break each cycle of $CC$ by deleting an edge that goes from a string to another with equal or larger period.

Now,
$$OV \leq \frac{2}{3} \sum_{c \in CC} p(t_i) = \frac{2}{3} \text{CYC}(G_S) \leq \frac{2}{3} \text{opt}(S).$$
And,
$$|\epsilon| = \text{CYC}(G_T) + OV \leq \frac{2}{3} \text{opt}(S).$$

The Second Improvement

O-R Lemma II: $\forall a \exists k \forall s \hspace{0.5cm} ov(s, a[k]) \leq p(s) + \frac{1}{2} p(\alpha)$.

Lemma: If $\text{apx}(T)$ is the length of the superstring of $T$ produced by some $\delta$ overlap approximation algorithm, then,
$$\text{apx}(T) \leq \sum_{t \in T} |t_i| - \delta \text{ maxov}(T) = \text{opt}(T) + (1 - \delta) \text{ maxov}(T).$$

Lemma: Assume that a shortest superstring of $T$ is $(t_1, \ldots, t_r)$. Then,
$$\text{maxov}(T) = \sum_{i=1}^{r-1} ov(t_i, t_{i+1}) \leq \sum_{i=1}^{r} \frac{3}{2} p(t_i) = \frac{3}{2} \text{CYC}(G_S).$$

Consequence: Using the $\frac{38}{63}$ overlap approximation algorithm of [KPS94]:
$$\text{apx}(T) \leq \text{opt}(T) + (1 - \frac{38}{63}) \text{ maxov}(T) \leq 2 \text{opt}(S) + \frac{26}{63} \cdot \frac{3}{2} \text{CYC}(G_S) \leq 2 \frac{25}{42} \text{opt}(S) \approx 2.596 \text{opt}(S).$$
Proof of the O-R Lemma (I)

∀ε∃k∈N, s inequivalent to α and p(s) ≤ p(α),
\[ o_v(s, \alpha[k]) \leq \frac{2}{3}(p(s) + p(\alpha)) \].

A string w is unbordered if it has no proper prefix that is also a suffix. Namely, f(w) = w. E.g., ababb.

A non-trivial factorization \((u, v)\) of w is a non-empty prefix u and suffix v of w = uv.

The local factor of a factorization \((u, v)\) of w = uv is the shortest string x that is consistent with both sides of the factorization \((u, v)\).

Example:

\[
\begin{array}{c|ccc}
\text{a} & b & a & b \\
\text{b} & a & a & a & a \\
\text{a} & a & a & a & a \\
\end{array}
\]

(a) (b) (c)

A factorization \((u, v)\) of w = uv is critical if its local factor x has length \(p(w)\).

The Critical Factorization Theorem [CV78]:

Given any \(p(w) - 1\) consecutive non-trivial factorizations of w, at least one is a critical factorization.

Proof of the O-R Lemma (II)

Lemma:

Let w be unbordered and have critical factorization \((u, v)\). Then,

1. the rotation \(w' = uv\) is also unbordered; and
2. \((v, u)\) is a critical factorization of \(w'\).

Proof: by contradiction.

1. \[
\begin{array}{cccc}
u & v & u & u \\
u & v & u & u \\
u & v & u & u \\
u & v & u & u \\
u & v & u & u \\
\end{array}
\]

2. \[
\begin{array}{cccc}
u & v & u & u \\
u & v & u & u \\
u & v & u & u \\
u & v & u & u \\
u & v & u & u \\
\end{array}
\]

Open Problems

1. Is GREEDY a 2-approximation?
2. Find better polynomial time approximation algorithms for length or for overlap.