Report Number: WUCS-86-09

1986-04-01

The Complexity of the Shortest Common Matching String Problem

Jonathan S. Turner

This paper describes the shortest common matching string problem, which arises from a data analysis problem in molecular genetics, and shows that it is NP-complete.

THE COMPLEXITY OF THE SHORTEST COMMON MATCHING STRING PROBLEM

Jonathan S. Turner

WUCS-86-9

April 1986

Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
Saint Louis, MO 63130-4899

Abstract

This paper describes the *shortest common matching string problem*, which arises from a data analysis problem in molecular genetics, and shows that it is NP-complete.

This research described here was supported in part by the National Science Foundation (grant #DCR-8409435) and the National Institutes of Health (grant #RR-01380).

The Complexity of the Shortest Common Matching String Problem

Jonathan S. Turner

Let $s_1 = a_1 \ldots a_r$ and $s_2 = b_1 \ldots b_s$ be strings over some finite alphabet Σ. We say that s_1 is a substring of s_2 if there is an integer $i \in [0, s-r]$ such that $a_j = b_{i+j}$ for $1 \leq j \leq r$. We also say in this case that s_2 is a *superstring* of s_1.

A *bag* $b = \langle a_1, \ldots, a_r \rangle$ is an unordered collection of symbols from some alphabet Σ in which the same symbol may appear more than once. (A bag is often referred to as a *multi-set.*) If $s = b_1 \ldots b_s$ is a string we define $\langle s \rangle$ to be the bag $\langle b_1, \ldots, b_s \rangle$. We say that a bag b *matches* a string s if s contains some substring s' such that $\langle s' \rangle = b$. We also say that s matches b or that s is a *matching string* of b. For example, the string **debcabf** is a matching string of the bag $\langle \mathbf{a}, \mathbf{b}, \mathbf{b}, \mathbf{c} \rangle$.

An instance of the *shortest common matching string problem* (SCMS) is a set of bags $B = \{b_1, \ldots, b_n\}$ over a finite alphabet Σ and an integer m. The object of the problem is to determine if there is a string of length $\leq m$ that matches every bag in B. Alternatively, we can view the object as being to find a minimum length string that matches every bag in B. We let $\chi^*(B)$ denote the length of a minimum length matching string for B.

EXAMPLE. If $B = \{\langle \mathtt{aceghi} \rangle, \langle \mathtt{abfgik} \rangle, \langle \mathtt{adfhki} \rangle, \langle \mathtt{defghi} \rangle, \langle \mathtt{afghik} \rangle\}$, the string **bfgiakhfdegiach** is a minimum length solution.

This problem has applications to molecular genetics. In particular, it arises in the analysis of experimental data used to map restriction enzyme sites in DNA from complex organisms. This connection is explained fully in [4]. The problem does not appear to have been studied previously, although a related problem, the *shortest common superstring problem* (SCS) has been [1,2,3].

The purpose of this paper is to introduce the shortest common matching string problem and prove that it is NP-complete. The transformation is from the shortest common superstring problem [1]. To simplify the presentation, we introduce an intermediate problem and use a two step transformation from SCMS to SCS.

Let $s = a_1 \ldots a_r$ be a string. The notation $s[i]$ denotes the symbol a_i if $i > 0$ and a_{r+i+1} if $i < 0$. The notation $s[i,j]$ denotes the string $s[i] \ldots s[j]$.

If $s = a_1 \ldots a_r$ is a string, we let $rev(s) = a_r \ldots a_1$.

The number of symbols in a string s is denoted $|s|$ and for any set of strings S, $\|S\| = \sum_{s \in S} |s|$.

An instance of the *shortest common superstring problem* is a set of strings $S = \{s_1, \ldots, s_n\}$ over a finite alphabet Σ and an integer m. The object of the problem is to determine if there is a string of length $\leq m$ that is a superstring of every $s_i \in S$.

EXAMPLE. If $S = \{\texttt{egiach}, \texttt{bfgiak}, \texttt{hfdegi}, \texttt{iakhfd}, \texttt{fgiakh}\}$, the string **bfgiakhfdegiach** is a minimum length solution.

The NP-completeness of SCS is proved in reference [1].

An instance of the *reversible shortest common superstring problem* is also a set of strings $S = \{s_1, \ldots, s_n\}$ over a finite alphabet Σ and an integer m. The object in this case, is to determine if there a string of length $\leq m$ that for all $s \in S$ contains either s or $rev(s)$.

EXAMPLE. If $S = \{\texttt{hcaige}, \texttt{kaigfb}, \texttt{igedfh}, \texttt{iakhfd}, \texttt{fgiakh}\}$, the string **bfgiakhfdegiach** is a minimum length solution.

THEOREM 1. RSCS *is* NP*-complete.*

Proof. Clearly RSCS $\in$ NP since a nondeterministic Turing machine can guess a string of length $\leq m$ and check in polynomial time that it is a superstring of either s or $rev(s)$ for all $s \in S$.

We now show how to transform an instance $(S = \{s_1, \ldots, s_n\}, \Sigma, m)$ of SCS to an instance $(S' = \{s'_1, \ldots, s'_n\}, \Sigma', m')$ of RSCS. We assume without loss of generality that no string in S is a substring of another.

First, define $\Sigma' = \Sigma \cup \{0, 1\}$ where 0 and 1 are not in Σ. For any string $s = a_1 \ldots a_r$, define $f(s) = 0a_1 10a_2 1 \cdots 0a_r 10$. We now define $s'_i = f(s_i)$ and let $m' = 3m + 1$.

For example, if $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}\}$, $S = \{$dccbda, bacbad, bdaabc, cbadcc$\}$ and $m = 14$ then $\Sigma' = \{\mathtt{a}, \mathtt{b}, \mathtt{c}, \mathtt{d}, 0, 1\}$, $m' = 43$ and

$$S' = \{ \texttt{0d10c10c10b10d10a10}, \texttt{0b10a10c10b10a10d10}, \texttt{0b10d10a10a10b10c10}, \texttt{0c10b10a10d10c10c10} \}$$

The original problem has the string **bacbadccbdaabc** as a solution. The corresponding solution to the transformed problem is

0b10a10c10b10a10d10c10c10b10d10a10a10b10c10

We claim that in general S has a solution of size $\leq m$ if and only if S' has a solution of size $\leq m'$. First, assume that σ is a superstring of all $s_i \in S$ and that $|\sigma| \leq m$. Renumber the s_i in order of their first appearance in σ and let π_i be the smallest j such that $s_i = \sigma[j, j+|s_i|-1]$. We will assume without loss of generality that $\pi_1 = 1$, $\pi_{i+1} \leq \pi_i + |s_i|$ for $1 \leq i \leq n-1$ and $\pi_n = |\sigma| - |s_n| + 1$. Now, let $\psi_i = \pi_i + |s_i| - \pi_{i+1}$ for $1 \leq i \leq n-1$ be the amount of overlap between consecutive strings in σ and note that for $1 \leq i \leq n-1$, $s'_i[-(3\psi_i + 1), -1] = s'_{i+1}[1, (3\psi_i + 1)]$. Hence, the string

$$\sigma' = s'_1 s'_2[3\psi_1 + 1, -1] s'_3[3\psi_2 + 1, -1] \cdots s'_n[3\psi_{n-1} + 1, -1]$$

is a superstring of all strings in S' and $|\sigma'| = 3|\sigma| + 1 \leq 3m + 1 = m'$. Hence, if S has a solution of size $\leq m$, S' has a solution of size $\leq m'$.

We now show that if S' has a solution of size $\leq m'$, then S must have a solution of size $\leq m$. Let σ' be any string which for all $s'_i \in S'$ is a superstring of either s'_i or $rev(s'_i)$ and let $|\sigma'| \leq m'$. Renumber the s_i in order of the first appearance of either s_i or $rev(s_i)$ in σ' and let π'_i be the smallest j such that either $s'_i = \sigma[j, j+|s'_i|-1]$ or $rev(s'_i) = \sigma[j, j+|s'_i|-1]$. We will assume without loss of generality that $\pi'_1 = 1$, $\pi'_{i+1} \leq \pi'_i + |s'_i|$ for $1 \leq i \leq n-1$ and $\pi'_n = |\sigma'| - |s'_n| + 1$. Now, let $\psi'_i = \pi'_i + |s'_i| - \pi'_{i+1}$ for $1 \leq i \leq n-1$ be the amount of overlap between consecutive strings. Note that if $\psi'_i > 1$ then either $\sigma'[\pi'_i + 2, \pi'_i + 3] = \sigma'[\pi'_{i+1} + 2, \pi'_{i+1} + 3] = 10$ or $\sigma'[\pi'_i, \pi'_i + 1] = \sigma'[\pi'_{i+1}, \pi'_{i+1} + 1] = 01$. That is, either both s'_i and s'_{i+1} are reversed in σ' or neither one is. Consequently, there is a string σ'' of the same length as σ' which is a superstring of all the strings in S' (that is, none of the strings is reversed in σ''). We will assume therefore, that each $s' \in S$ is a substring of σ'. Hence

$$\sigma = s_1 s_2[((\psi'_1 - 1)/3) + 1, -1] s_3[((\psi'_2 - 1)/3) + 1, -1] \cdots s_n[((\psi_{n-1} - 1)/3) + 1, -1]$$

is a superstring of all strings in S and $|\sigma| = (|\sigma'| - 1)/3 \leq (m' - 1)/3 = m$. To complete the proof, we note that (S', Σ', m') can be computed deterministically in time polynomial in $\|S\|$. □

Remark. In [1], it is shown that SCS is NP-complete, even when the alphabet is limited to two symbols. Since the proof of Theorem 1 adds just two symbols to the alphabet, it follows that RSCS is NP-complete when the alphabet is limited to four symbols.

THEOREM 2. *SCMS is NP-complete.*

Proof. Clearly SCMS is in NP, since a nondeterministic Turing machine can guess a string of length $\leq m$ and check in polynomial time that it matches each of the bags.

We now show how to transform an instance $(S = \{s_1, \ldots, s_n\}, \Sigma, m)$ of RSCS to an instance $(B = \{b_1, \ldots, b_p\}, \Sigma', m')$ of SCMS, where $\Sigma' = \Sigma$ together with the new symbols $\{\mathrm{L}, \mathrm{R}, \mathrm{x}_1, \ldots, \mathrm{x}_n\}$, $m' = n(r+2) + m + \|S\|$ and $r = 1 + 4\|S\|$. $B = B_1 \cup \cdots \cup B_n$ where

$$\begin{aligned} B_i &= \{\langle \mathrm{x}_i^r \rangle, \langle \mathrm{L}\mathrm{x}_i^r \rangle, \langle \mathrm{x}_i^r \mathrm{R} \rangle\} \cup \{\langle \mathrm{x}_i^r \mathrm{R} s_i[1,j] \rangle \mid 1 \leq j \leq |s_i|\} \\ &\quad \cup \{\langle s_i[j,-1] \mathrm{L}\mathrm{x}_i^r \rangle \mid 1 \leq j \leq |s_i|\} \end{aligned}$$

For example, if $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$, $S = \{\mathrm{bcdb}, \mathrm{dcbc}, \mathrm{abcb}\}$ and $m = 7$ then $\Sigma' = \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}, \mathrm{L}, \mathrm{R}, \mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3\}$, $m' = 172$ and $B = B_1 \cup B_2 \cup B_3$ where

$$\begin{aligned} B_1 = \{ \; & \langle \mathrm{x}_1^{49} \rangle, \langle \mathrm{L}\mathrm{x}_1^{49} \rangle, \langle \mathrm{x}_1^{49}\mathrm{R} \rangle, \langle \mathrm{x}_1^{49}\mathrm{Rb} \rangle, \langle \mathrm{x}_1^{49}\mathrm{Rbc} \rangle, \langle \mathrm{x}_1^{49}\mathrm{Rbcd} \rangle, \langle \mathrm{x}_1^{49}\mathrm{Rbcdb} \rangle, \\ & \langle \mathrm{bL}\mathrm{x}_1^{49} \rangle, \langle \mathrm{dbL}\mathrm{x}_1^{49} \rangle, \langle \mathrm{cdbL}\mathrm{x}_1^{49} \rangle, \langle \mathrm{bcdbL}\mathrm{x}_1^{49} \rangle \} \end{aligned}$$

The sets B_2 and B_3 are similar. The original problem has the string abcbcdb as a solution. The corresponding solution to the transformed problem is

$$\mathrm{abcbL}\mathrm{x}_3^{49}\mathrm{Rabcbcd}\mathrm{R}\mathrm{x}_2^{49}\mathrm{Lcbcdb}\mathrm{L}\mathrm{x}_1^{49}\mathrm{Rbcdb}$$

Note that its length is 172.

We claim that in general S has a solution string of length $\leq m$ if and only if B has a solution string of length $\leq m'$. First, assume that σ is a superstring of either s_i or $rev(s_i)$ for all $s_i \in S$ and that $|\sigma| \leq m$. Renumber the s_i in order of the first appearance of either s_i or $rev(s_i)$ in σ and let π_i be the smallest j such that either $s_i = \sigma[j, j + |s_i| - 1]$ or $rev(s_i) = \sigma[j, j + |s_i| - 1]$. We will assume without loss of generality that $\pi_1 = 1$, $\pi_{i+1} \leq \pi_i + |s_i|$ for $1 \leq i \leq n-1$ and $\pi_n = |\sigma| - |s_n| + 1$. Define

$$s_i' = \begin{cases} s_i \mathrm{L}\mathrm{x}_i^r \mathrm{R} s_i & \text{if } \sigma[\pi_i, \pi_i + |s_i| - 1] = s_i \\ rev(s_i)\mathrm{R}\mathrm{x}_i^r \mathrm{L} rev(s_i) & \text{if } \sigma[\pi_i, \pi_i + |s_i| - 1] = rev(s_i) \end{cases}$$

for $1 \leq i \leq n$ and note that s_i' is a matching string for all the bags in B_i. Now, let $\psi_i = \pi_i + |s_i| - \pi_{i+1}$ for $1 \leq i \leq n-1$ be the amount of overlap between consecutive strings in σ and note that the string

$$\sigma' = s_1' s_2'[\psi_1 + 1, -1] s_3'[\psi_2 + 1, -1] \cdots s_n'[\psi_{n-1} + 1, -1]$$

is a matching string for all the bags in B and

$$|\sigma'| = n(r+2) + 2\|S\| - \sum_{i=1}^{n-1} \psi_i \leq n(r+2) + 2\|S\| - (\|S\| - m) = n(r+2) + m + \|S\| = m'$$

Hence, if S has a solution of length $\leq m$, then B has a solution of length $\leq m'$.

We now show that if B has a solution of length $\leq m'$ then S has a solution of length $\leq m$. Let σ' be a shortest matching string for B and assume that $|\sigma'| \leq m'$. Let π'_i be the smallest j such that $b_i = \langle \sigma'[j, j + |b_i| - 1] \rangle$. Define

$$\psi'_{ij} = |\{\pi'_i, \ldots, \pi'_i + |b_i| - 1\} \cap \{\pi'_j, \ldots, \pi'_j + |b_j| - 1\}|$$

That is, ψ'_{ij} is the amount of overlap between bags b_i and b_j in σ'.

Now, note that $nr \leq |\sigma'| \leq m' < (n+1)r$. Consequently for any $h \in [1, n]$ if $b_i = \langle \mathrm{x}^r_h \rangle$ and $b_j = \langle \mathrm{Lx}^r_h \rangle$ then $\psi'_{ij} \geq 1$. If $\pi'_j < \pi'_i - 1$ then the string $\sigma'[\pi'_j, \pi'_i]$ has the form $\mathrm{x}^s_h \mathrm{Lx}^t_h$, where $0 \leq s \leq \pi'_i - \pi'_j - 1$ and $s + t = \pi'_i - \pi'_j$. Hence, $\sigma'[1, \pi'_j - 1]\mathrm{L}\sigma'[\pi'_i, |\sigma'|]$ is also a matching string of B and is shorter that σ'. Since we assumed that σ' was a shortest matching string for B it follows that $\pi'_j \geq \pi'_i - 1$. Similarly, we can show that $\pi'_j \leq \pi'_i$ and consequently $\psi'_{ij} = r$. This argument can be extended to show that $\psi'_{ij} = r$ for any string $b_j \in B_h$ and $b_i = \langle \mathrm{x}^r_h \rangle$.

The above observations imply that for all $h \in [1, n]$, σ' contains a string s'_h of the form $s_h \mathrm{Lx}^r_h \mathrm{R} s_h$ or $rev(s_h)\mathrm{Rx}^r_h \mathrm{L} rev(s_h)$. Renumber the s'_h in order of their first appearance in σ' and note that for all $i \in [1, n-1]$ the overlapping portion of s'_i and s'_{i+1} is also a valid overlap for s_i and s_{i+1}. Now, redefine π'_i to be the smallest j such that $\sigma'[j, j + |s'_i| - 1] \in \{s'_i, rev(s'_i)\}$ for $1 \leq i \leq n$, and redefine $\psi'_i = \pi'_i + |s'_i| - \pi'_{i+1}$ for $1 \leq i \leq n-1$ and note that the string

$$\sigma = s_1 s_2[\psi'_1 + 1, -1] s_3[\psi'_2 + 1, -1] \ldots s_n[\psi'_{n-1} + 1, -1]$$

is a solution to the original RSCS instance and that

$$|\sigma| = \|S\| - \sum_{i=1}^{n-1} \psi'_i \leq \|S\| - \left[\sum_{i=1}^{n} |s'_i| - m'\right] = m' - n(r+2) - \|S\| = m$$

Hence, whenever B has a solution of length $\leq m'$, S has a solution of length $\leq m$. To complete the proof, we note that (B, Σ', m') can be computed deterministically in time polynomial in $\|S\|$. □

The NP-completeness of SCMS makes it unlikely that there exists an efficient algorithm to solve it exactly. In a separate paper [5] we address the issue of good approximation algorithms.

References

[1] Gallant, John K. "On finding Minimal Length Superstrings," *Journal of Computer and System Sciences*, vol. 20, no. 1, 50–58, 2/80.

[2] Gallant, John K. "String Compression Algorithms," Ph.D. Dissertation, Princeton University, Department of Electrical Engineering and Computer Science, June 1982.

[3] Maier, David and James A. Storer. "A Note on the Complexity of the Superstring Problem," Princeton University Technical Report 233, Department of Electrical Engineering and Computer Science, October 1977.

[4] Turner, Jonathan S. "The DNA Mapping Problem," Washington University, Computer Science Department technical report, in preparation.

[5] Turner, Jonathan S. "Algorithms for the Shortest Common Matching String Problem," Washington University, Computer Science Department technical report, in preparation.