A W[2]-Hard Variant of Common Approximate Substring

Andrew D. Smith
Faculty of Computer Science,
University of New Brunswick
Fredericton, NB, Canada,
p7ka@unb.ca

August 14, 2002

Abstract

Given a set $\mathcal{F}$ of strings and two integers $d < l$, the common approximate substring problem asks whether there is a string $\mathcal{C}$ of length l, such that each member of $\mathcal{F}$ contains a substring that differs from $\mathcal{C}$ in at most d positions. For a fixed number of strings, and fixed alphabet size, Fellows et al. [2] showed this problem to be hard for the class *W[1]* of the *W*-hierarchy. We strengthen this result and show that the problem is *W[2]*-hard by a parameterized reduction from set cover.

1 Introduction

This paper examines the COMMON APPROXIMATE SUBSTRING problem, formally defined as follows:

COMMON APPROXIMATE SUBSTRING (CAS)

Instance:	A set $\mathcal{F} = \{S_1, \ldots, S_m\}$ of strings over an alphabet Σ such that $\|S_i\| = n$, $1 \leq i \leq m$, and positive integers l and d such that $1 \leq d \leq l \leq n$.
Parameter:	Alphabet Σ and positive integers m, n, l and d.
Question:	Is there a string $\mathcal{C} \in \Sigma^l$ such that for each string $S \in \mathcal{F}$, $\mathcal{C}$ is Hamming distance $\leq d$ from some length-l substring of $\mathcal{F}$?

We call $\mathcal{C}$ the *center string* for $\mathcal{F}$. When comparing strings, we use the notation $d_H(a, b)$ to denote the Hamming distance between strings a and b. When $|a| < |b|$, $d_H(a, b) = \min_{b' \in b} d_H(a, b')$, where b' is a substring of b. The original NP-completeness proof for COMMON APPROXIMATE SUBSTRING was for the more restricted case with $l = n$ and $|\Sigma| = 2$ in [3]. This result of this paper both strengthens and complements the *W[1]*-hardness result given in [2] for CAS$(m, |\Sigma|)$ when $|\Sigma| = 2$.

Parameterized complexity analysis. Parameterized complexity provides general methods for obtaining exact solutions to *NP*-hard problems. A parameterized problem is a decision problem whose instances include one or more values that may be fixed, independent of the input size. An example of such a problem is k-vertex cover, where the size k of the cover is fixed, and does not depend on the size of the graph. A problem is said to be fixed parameter tractable if it can be solved by an algorithm with time complexity $O(f(k)n^c)$, where n is the size of the input, k is the parameter, f is an arbitrary function that does not depend on n, and c is constant.

The theory of parameterized complexity also describes and organizes the complexity of problems that are not fixed parameter tractable. Parameterized complexity draws fine distinctions between *NP*-hard problems and classifies them according to an infinite hierarchy of complexity classes within *NP*. The hierarchy is called the *W-hierarchy* $= \{W[1], W[2], \ldots, W[t]\}$, and is based on successively more powerful boolean circuits. One can establish that a parameterized problem Π is hard for a class in the *W*-hierarchy by using a parameterized reduction from a problem known to be a member of that class.

Definition 1 Let Π and Π' be two parameterized problems. A parameterized reduction from Π to Π' is an algorithm A that transforms an instance

$\langle x, k\rangle$ of Π into an instance of $\langle x', k'\rangle$ of Π' such that:

1. A runs in time $O(f(k)|x|^c)$ for arbitrary function f (independent of $|x|$) and constant c (independent of both x and k).
2. $k' = g(k)$ for some arbitrary function g independent of $|x|$.
3. $\langle x, k\rangle \in \Pi$ if and only if $\langle x', k'\rangle \in \Pi'$.

For this paper, the version of COMMON APPROXIMATE SUBSTRING parameterized with the alphabet size ($|\Sigma|$) and number of strings (m) is denoted CAS$(m, |\Sigma|)$. We have only mentioned those concepts of parameterized complexity that are essential to the material in this paper. A detailed description of the theory can be found in [1].

2 The W[2]-hardness of CAS$(m, |\Sigma|)$

To show *W[2]*-hardness for CAS$(m, |\Sigma|)$, we give a parameterized reduction from the *W[2]*-complete problem SET COVER [1].

SET COVER [4, Problem SP5]

Instance:	A set $\mathcal{B}$ of elements, a family of sets $\mathcal{L}$ such that $\mathcal{L}_i \subseteq \mathcal{B}, (1 \leq i \leq \|\mathcal{L}\|)$ and a positive integer k.
Parameter:	A positive integer k.
Question:	Is there a size k subset $R \subseteq \mathcal{L}$ such that $\cup_{R_j \in R} R_j = \mathcal{B}$?

Let $I = \langle \mathcal{B}, \mathcal{L}\rangle$ be an instance of SET COVER. Without loss of generality, assume that the elements of $\mathcal{B}$ are the integers $[1, |\mathcal{B}|]$. We show how to construct an instance $\mathcal{F}$ of CAS$(m, |\Sigma|)$ such that I has a cover of size k if and only if $\mathcal{F}$ has a center with a particular maximum distance to any instance.

Target Parameters. The number of strings in $\mathcal{F}$ is $m = f_1(k) = 2k$. The length of the center $\mathcal{C}$ is $l = f_2(\mathcal{B}, k) = k|\mathcal{B}| + 2$, and the maximum distance between instance and center is $d = f_3(\mathcal{B}, k) = (k-1)|\mathcal{B}|$. The maximum length of the strings in $\mathcal{F}$ is $n = f_4(\mathcal{L}, \mathcal{B}, k) = 2(k|\mathcal{B}| + 2) \cdot |\mathcal{L}|$, and the alphabet size is $|\Sigma| = f_5(k) = 3k + 1$.

The Alphabet. The string alphabet is $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \{A\}$. We refer to these as *solution characters* (Σ_1), *unique characters* (Σ_2) and the *alignment character* (A), with

$$\begin{aligned} \Sigma_1 &= \{s_1, \ldots, s_k\}, \\ \Sigma_2 &= \{u_{11}, u_{12}, u_{21}, u_{22}, \ldots, u_{k1}, u_{k2}\}. \end{aligned}$$

For $1 \leq i \leq k$, we will assume without loss of generality that character s_i is the integer i. The characters of Σ_2, denoted by u with subscripts, are identical within a string, but different between strings.

Substring Gadgets. We next describe the three "high level" component substrings used in the construction. For Membership Indicators, the product symbol refers to concatenation and is ordered.

Fillers:

$$\langle Cover(i) \rangle = s_i^{(k-1)|\mathcal{B}}$$

Separators:

$$\langle Separator(i,p) \rangle = u_{ip}^{(k|\mathcal{B}|+2)}$$

Member Indicators:

$$\langle Set(i,j,p) \rangle = \prod_{b \in \mathcal{B}} g(i,j,p,b)$$

The Cover Indicators are strings of length $(k-1)|\mathcal{B}|$ and each corresponds to some $\mathcal{L}_i \in \mathcal{L}$. The Separators are strings of length $k|\mathcal{B}|+2$. Each is composed entirely of characters from Σ_2, and the variable p takes values from $\{1,2\}$. The Set Indicators are used to indicate the sets that make up a cover. The function g is defined as

$$g(i,j,p,b) = \begin{cases} s_i & \text{if } \ b \in \mathcal{L}_j, \\ u_{ip} & \text{otherwise.} \end{cases}$$

The Reduction. Each of the k sets in the cover R for I is represented by a pair of strings in $\mathcal{F}$. In particular, the instances in strings $S_{i1}, S_{i2} \in \mathcal{F}$ correspond to the i^{th} set in R. Define

$$S_{i1} = \prod_{1 \leq j \leq |\mathcal{L}|} A\langle Set(i,j,1) \rangle \langle Cover(i) \rangle A \langle Separator(i,1) \rangle,$$

$\mathcal{B}$:	$\{1, 2, 3, 4, 5, 6, 7\}$
$\mathcal{L}_1$:	$\{1, 4, 6\}$
$\mathcal{L}_2$:	$\{1, 2, 4\}$
$\mathcal{L}_3$:	$\{3, 5\}$
$\mathcal{L}_4$:	$\{1, 2, 3, 7\}$
k:	3

Figure 1: Example instance of SET COVER.

S_{11} : A1uu1u1u1^{14}Au^{23}A11u1uuu1^{14}Au^{23}Auu1u1uu1^{14}Au^{23}A111uuu11^{14}Au^{23}A
S_{12} : A1uu1u1u1^{14}Au^{23}A11u1uuu1^{14}Au^{23}Auu1u1uu1^{14}Au^{23}A111uuu11^{14}Au^{23}A
S_{21} : A2uu2u2u2^{14}Au^{23}A22u2uuu2^{14}Au^{23}Auu2u2uu2^{14}Au^{23}A222uuu22^{14}Au^{23}A
S_{22} : A2uu2u2u2^{14}Au^{23}A22u2uuu2^{14}Au^{23}Auu2u2uu2^{14}Au^{23}A222uuu22^{14}Au^{23}A
S_{31} : A3uu3u3u3^{14}Au^{23}A33u3uuu3^{14}Au^{23}Auu3u3uu3^{14}Au^{23}A333uuu33^{14}Au^{23}A
S_{32} : A3uu3u3u3^{14}Au^{23}A33u3uuu3^{14}Au^{23}Auu3u3uu3^{14}Au^{23}A333uuu33^{14}Au^{23}A

Figure 2: CAS$(m, |\Sigma|)$ representation for the example instance of SET COVER given in Figure 5. The character u denotes a character that differs across strings. The underlined substrings are expanded and explained in Figure 3.

$$S_{i2} = \prod_{1 \le j \le |\mathcal{L}|} A\langle Set(i, j, 2)\rangle\langle Cover(i)\rangle A\langle Separator(i, 2)\rangle.$$

The family of strings is then $\mathcal{F} = \{S_{11}, S_{12}, S_{21}, S_{22}, \ldots, S_{k1}, S_{k2}\}$. Note that no matter what set of substrings is taken as instances of a center, aside from the positions containing alignment characters, any position has at most two strings with the same character. An example of this reduction is provided in Figures 1, 2, and 3. For Figure 2, the subscripts are left out of the unique characters; these are given unique symbols in Figure 3.

The proof of correctness for this reduction rests on a function $\hat{d}$ that provides a lower bound on $\overline{d}$, the minimum possible value of d for a set of instances. Given a collection of potential instances of a center string for $\mathcal{F}$, define z_{ij} as the indicator function that has the value 1 if $S_j[i]$ is *not* the column majority character in column i of the aligned instances and the value

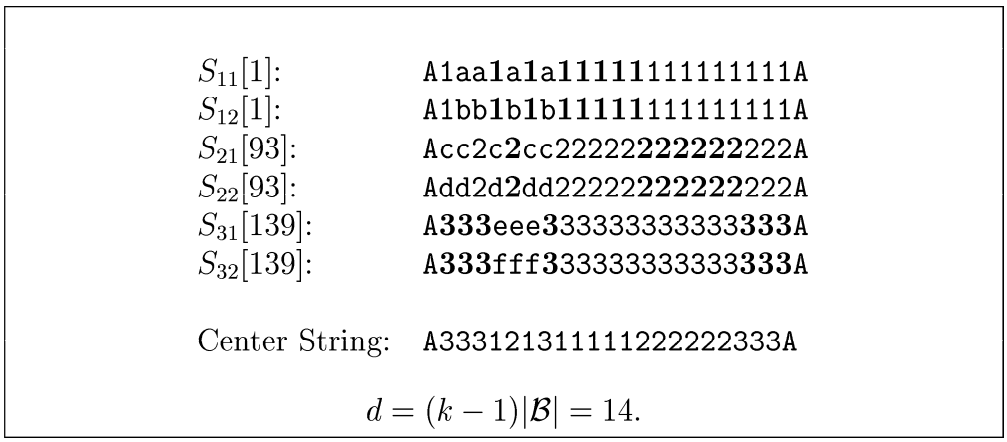

Figure 3: Expanded diagrams of the substrings underlined in Figure 2, along with an optimal center string for the instances. The characters in bold are those that match the center string.

0 otherwise. The function $\hat{d}$ is defined as follows:

$$\hat{d} = \left\lceil \frac{\Sigma_{i=1}^{n} \Sigma_{j=1}^{m} z_{ij}}{m} \right\rceil.$$

Intuitively, $\hat{d}$ counts the number of mismatches occuring between the instances and their center in the best case, then distributes them evenly among the instances. Those rows for which the distance to the center is not less than d are referred to as *bad rows*, and their number is denoted by r.

Lemma 1 $\hat{d} \leq \overline{d}$.

Proof 1 We proceed by induction on the number of columns considered. This induction will use the functions $\overline{d}_h$ and $\hat{d}_h$ which denote the values of $\overline{d}$ and $\hat{d}$ relative to the first h characters of the given instance strings. The base case is the trivial situation of a single column which can easily be verified. For the inductive step, suppose $\hat{d}_h \leq \overline{d}_h$. If $\hat{d}_h < \overline{d}_h$ or if $\overline{d}_h < \overline{d}_{h+1}$, the addition of a column cannot increase $\hat{d}_{h+1}$ beyond $\overline{d}_{h+1}$. It remains to establish the case where $\hat{d}_h = \overline{d}_h = \overline{d}_{h+1}$. Since this case does not see an increase in $\overline{d}$, it must be true that all bad rows have the same character in column $h+1$. This implies the following:

$$
\begin{aligned}
\hat{d}_{h+1} &= \left\lceil \frac{\Sigma_{i=1}^{h+1}\Sigma_{j=1}^{m} z_{ij}}{m} \right\rceil \\
&= \left\lceil \frac{\Sigma_{i=1}^{h}\Sigma_{j=1}^{m} z_{ij}}{m} + \frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m} \right\rceil \\
&= \left\lceil \frac{\sum_{i=1}^{h}\left(z_{i1} + \ldots + z_{im}\right)}{m} + \frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m} \right\rceil \\
&= \left\lceil \frac{\left(z_{11} + \ldots + z_{1m}\right) + \ldots + \left(z_{h1} + \ldots + z_{hm}\right)}{m} + \frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m} \right\rceil \\
&= \left\lceil \frac{\left(z_{11} + \ldots + z_{h1}\right) + \ldots + \left(z_{1m} + \ldots + z_{hm}\right)}{m} + \frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m} \right\rceil .
\end{aligned}
$$

Given this expansion and reordering of z_{ij} terms by row, note that there will be r bad rows taking value 1, so we simply remove r of these z_{ij}, one for each of the bad rows, and we are left with $\overline{d}_h - 1$ of the z_{ij} that take a value of 1 in each of the m rows, *i.e.*,

$$
\frac{\sum_{i=1}^{h}\sum_{j=1}^{m} z_{ij}}{m} = \overline{d}_h - 1 + \frac{r}{m}.
$$

This in turn implies:

$$
\hat{d}_{h+1} = \left\lceil \overline{d}_h - 1 + \frac{r}{m} + \frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m} \right\rceil = \overline{d}_h - 1 + \left\lceil \frac{r}{m} + \frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m} \right\rceil .
$$

Observe that if $\overline{d}_{h+1} = \overline{d}_h$, then all bad rows have the same character in column $h+1$. Therefore the value of the term $\frac{\Sigma_{j=1}^{m} z_{h+1,j}}{m}$ is at most $\left(1 - \frac{r}{m}\right)$, and so

$$
\hat{d}_{h+1} = \overline{d}_h - 1 + \left\lceil \frac{r}{m} + \left(1 - \frac{r}{m}\right) \right\rceil = \overline{d}_h .
$$

∎

Lemma 2 *If the column majority character occurs at most twice in any column, then* $\overline{d} \geq l - \frac{2l}{m}$.

Proof 2 Suppose no column contains any character more than twice. By Lemma 1,

$$\overline{d} \geq \hat{d} \geq \left\lceil \frac{(m-2)l}{m} \right\rceil .$$

∎

Lemma 3 *Let $\mathcal{F}$ be a set of strings constructed as described in the reduction and let $\mathcal{C}$ be a center for $\mathcal{F}$. Then $\mathcal{C}$ must begin and end with the alignment character, and so must all instances.*

Proof 3 Suppose the center $\mathcal{C}$ does not begin with the alignment character; then by the separation between alignment characters in members of $\mathcal{F}$, no instance can match two alignment characters in $\mathcal{C}$. As all but one column has at most two occurrences of the column majority character, we can rewrite the bound on $\overline{d}$ given in Lemma 2 with the substitutions $m = 2k$ and $l = k|\mathcal{B}|+2$ to obtain $\overline{d} \geq (k-1)|\mathcal{B}| + \frac{k-1}{k}$. A symmetric argument establishes that $\mathcal{C}$ must end with the alignment character.

Suppose some instance begins or ends with a character other than the alignment character. Then that instance cannot match $\mathcal{C}$ at those positions. Again using Lemma 2, $\overline{d} \geq (k-1)|\mathcal{B}| + \frac{k-1}{k}$. ∎

Lemma 4 SET COVER *parametrically reduces to* CAS$(m, |\Sigma|)$.

Proof 4 The construction described above runs in time that is fixed-parameter tractable relative to m and $|\Sigma|$. Hence, we need only show that the reduction is correct.

Suppose there is a cover R for $\mathcal{B}$, such that $|R| = k$. From R, construct a center $\mathcal{C}$ for $\mathcal{F}$ as follows. (1) The first and last positions of $\mathcal{C}$ are assigned the alignment character A. (2) The next $|\mathcal{B}|$ positions, used to represent elements of the base set, are each assigned a character indicating one of the sets in R that covers the corresponding element. For each $b \in \mathcal{B}$, choose some $\mathcal{L}_i \in R$, such that $b \in \mathcal{L}_i$, as covering b. Since R is a cover for $\mathcal{B}$, there is at least one such choice for every $b \in \mathcal{B}$. If $\mathcal{L}_i$ is chosen to cover b, then s_i is assigned to position $b+1$ in $\mathcal{C}$ (recall that the elements of $\mathcal{B}$ have been equated with the integers 1 to $|\mathcal{B}|$). (3) The remaining $(k-1)|\mathcal{B}|$ positions of $\mathcal{C}$ correspond to the Filler gadgets. For each $\mathcal{L}_i \in R$, if x_i positions ($0 \leq x_i \leq |\mathcal{B}|$) of $\mathcal{C}$ have been assigned characters corresponding to elements in $\mathcal{L}_i$, then $|\mathcal{B}| - x_i$ positions in the Filler part of $\mathcal{C}$ are assigned the character s_i. Given this

construction, note that if $\mathcal{L}_j \in \mathcal{L}$ is the i^{th} set in R, then the substring of S_{i1} (and of S_{i2}) that begins and ends with the alignment character, and contains the j^{th} Membership Indicator, matches $\mathcal{C}$ in exactly $2+|\mathcal{B}|$ positions. Therefore $\mathcal{C}$ is a center for $\mathcal{F}$ with distance exactly $(k-1)|\mathcal{B}|$ to any instance.

Conversely, suppose there is no cover of size k for $\mathcal{B}$. Then for any set of instances from members of $\mathcal{F}$, there exists at least one position in which every instance has a unique character. Lemma 2 provides a lower bound on the d value of any center for the instances. Any position j will contribute $\sum_{i=1}^{m} z_{ij} = (m-2)$ to $\hat{d}$, except for those columns having a unique character in each instance, which contribute $m-1$. This implies the following:

$$\begin{aligned} \hat{d} &\geq \left\lceil \frac{(l-3)(m-2)+(m-1)}{m} \right\rceil \\ &= \left\lceil k|\mathcal{B}| + \frac{5}{2k} - \frac{2(k|\mathcal{B}|+2)}{2k} \right\rceil \\ &\geq (k-1)|\mathcal{B}|+1. \end{aligned}$$

Therefore $\mathcal{F}$ has no center with maximum distance $(k-1)|\mathcal{B}|$. ∎

Theorem 1 CAS$(m, |\Sigma|)$ *is* W[2]*-hard.*

Proof 5 Follows from Lemma 4 and the *W[2]*-hardness of SET COVER [1]. ∎

References

[1] R. Downey and M. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer-Verlag, New York, 1999.

[2] M.R. Fellows, J. Gramm, and R. Niedermeier. On the parameterized intractability of closest substring and related problems. In H. Alt and A. Ferreira, editors, *The 19th International Symposium on Theoretical Aspects of Computer Science STACS 2002*, volume 2285 of *Lecture Notes in Computer Science*, pages 262–273, Antibes/Juan-Les-Pins, France, 2002. Springer.

[3] M. Frances and A. Litman. On covering problems of codes. *Theory of Computing Systems*, 30(2):113–119, 1997.

[4] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and Company; San Francisco, 1979.