Fast algorithm for two-dimensional pattern matching with k mismatches

Jonas Ellert[1], Paweł Gawrychowski[2], Adam Górkiewicz[3], and Tatiana Starikovskaya[4]

[1]?
[2]?
[3]?
[4]?

Abstract

1 Introduction

We consider the one-dimensional all-substring Hamming distance problem (HD1D), where for a given text string T of length n and a string P of length m ($m < n$), we want to calculate the Hamming distance between P and every fragment T of length m.

We consider the two-dimensional all-substring Hamming distance problem (HD2D), where for a given 2D string T of size $n \times n$ and a string P of size $m \times m$ ($m < n$), we want to calculate the Hamming distance between P and every $m \times m$ fragment of T.

We also consider the bounded variants of HD1D and HD2D, where we are only required to calculate the distances which are not greater than k, for some parameter $k \in \mathbb{Z}^+$.

Theorem 1 (Main result)**.** *Bounded* HD2D *can be solved in* $\tilde{O}((m^2 + mk^{5/4})n^2/m^2)$ *time.*

2 Preliminaries

For our purposes we will not use the standard definition of a two-dimensional string, where we associate it with a two-dimensional array of characters, and instead we will define it more broadly. Although we will occasionally use the array notation, we will do it exclusively for $n \times m$ strings. For any $n \in \mathbb{Z}^+$ we will denote $[n] = \{ 0, \dots, n-1 \}$. We will only consider integer points or vectors and we will use these terms interchangeably. For a function f and a subset of its domain X we will write $f \restriction_X$ to denote the restriction of f to X. Our results hold under word-RAM model of computation.

Definition 1 (Two-dimensional string)**.** We define a **string** S as a partial function $\mathbb{Z}^2 \nrightarrow \Sigma$ which maps some arbitrary set of integer points, denoted as $\mathrm{dom}(S)$, to characters. For simplicity we will write $u \in S$ to denote that $u \in \mathrm{dom}(S)$. We say that a string S is **partitioned** into strings $R_1, \dots, R_\ell$ when the sets $\mathrm{dom}(R_1), \dots, \mathrm{dom}(R_\ell)$ partition $\mathrm{dom}(S)$ and $R_i(u) = S(u)$ for all $u \in R_i$. We call a string S **monochromatic** when $S(u) = \sigma$ for every $u \in S$ for some $\sigma \in \Sigma$ and we will write $\mathrm{C}(S)$ to denote the value σ. We say that S is $n \times m$ for some $n, m \in \mathbb{Z}^+$ when $\mathrm{dom}(S) = [n] \times [m]$. Physically we represent a string as a list of point-character pairs.

Definition 2 (Shifting)**.** For a set of points $V \subseteq \mathbb{Z}^2$ and a vector $u \in \mathbb{Z}^2$, we denote $V + u$ as $\{ v + u : v \in V \}$. For a string S and a vector $u \in \mathbb{Z}^2$ we denote $S + u$ as a string R such that

$\mathrm{dom}(R) = \mathrm{dom}(S) + u$ and $R(v) = S(v - u)$ for $v \in \mathrm{dom}(R)$. Intuitively, we shift the set of points while maintaining their character values.

Definition 3 (Hamming distance)**.** For a pair of strings S, R we define

$$\mathrm{Ham}(S, R) = |\{ u : u \in \mathrm{dom}(S) \cap \mathrm{dom}(R), S(u) \neq R(u) \}|,$$

which corresponds to the number of mismatches between S and R.

Under such notation, the HD2D problem is equivalent to calculating the (bounded or unbounded) values of $\mathrm{Ham}(P + q, T)$ for all $q \in \mathbb{Z}^2$ such that $\mathrm{dom}(P + q) \subseteq \mathrm{dom}(T)$ (so for $q \in [n - m]^2$).

Definition 4 (Don't care symbol)**.** We define the **don't care** symbol as a special character which matches with every character. We will denote it with ?. Unless stated otherwise, we assume it is not allowed in Σ and in both HD1D and HD2D every character present in T and P matches only with itself.

Definition 5 (Vector operators)**.** For any $u \in \mathbb{Z}^2$ we refer to its coordinates as $u.x, u.y$. For $u, v \in \mathbb{Z}^2$ we denote $u \cdot v = u.x \cdot v.x + u.y \cdot v.y$ and $u \times v = u.x \cdot v.y - u.y \cdot v.x$. Note that alternatively $u \cdot v = |u||v| \cos \alpha$ and $u \times v = |u||v| \sin \alpha$ where α is the angle between u and v.

Definition 6 (Quadrants)**.** We define the four **quadrants** as

$$\begin{aligned}
\mathcal{Q}_1 &= (0, +\infty) \times [0, +\infty), \\
\mathcal{Q}_2 &= (-\infty, 0] \times (0, +\infty), \\
\mathcal{Q}_3 &= (-\infty, 0) \times (-\infty, 0], \\
\mathcal{Q}_4 &= [0, +\infty) \times (-\infty, 0).
\end{aligned}$$

3 One-dimensional generalizations

In this section we explore some of the methods used for one-dimensional strings. Specifically, as our goal is to generalize the solution for pattern matching with k mismatches described in [2], we are especially interested in two-dimensional variants of the techniques that were used to solve the one-dimensional case.

Theorem 2 (Instancing)**.** *Consider an algorithm $\mathcal{A}$ which solves* HD2D *(bounded or unbounded), but only when $2|n$ and $n \leq \frac{3}{2}m$. If its running time is $\mathcal{T}(m)$, then the general case can be solved in $\mathcal{O}(\mathcal{T}(m)n^2/m^2)$.*

Proof. Let $r = \lfloor m/2 \rfloor$ and let $n' = r + m - 1$ or $r + m$ if $r + m - 1$ is odd. We see that the set $N = [n']^2$ satisfies the conditions for the text domain. For any vector $q \in [n - m]^2$ we can find a vector u such that $r|u.x, r|u.y$ and $q - u \in [r]^2$, so we have $\mathrm{Ham}(P + q, T) = \mathrm{Ham}(P + q - u, T_u)$ where $T_u = (T - u) \restriction_N$. If $T - u$ is not defined for some $v \in N$, we can pad $T_u(v)$ with any character. We see that $\mathrm{dom}(P + q - u) \subseteq N = \mathrm{dom}(T_u)$. There are $\mathcal{O}(n^2/m^2)$ possible vectors u and we run $\mathcal{A}$ for every pair of T_u and P. ☐

Theorem 3 (Kangaroo jumps)**.** *Consider an $n \times n$ string T, $m \times m$ string P and set of vectors Q such that* $\mathrm{dom}(P + q) \subseteq \mathrm{dom}(T)$ *for every $q \in Q$. There exists an algorithm which calculates $d_q = \mathrm{Ham}(P + q, T)$ for every $q \in Q$ in total time $\tilde{\mathcal{O}}(n^2 + \sum_{q \in Q} d_q)$.*

Proof. For the sake of clarity, we will temporarily switch to the classical array notation for strings. Let $T_0, \ldots, T_{n-m}$ denote an array of two-dimensional strings (arrays) such that $T_k[0 \mathinner{.\,.} n - 1, 0 \mathinner{.\,.} m - 1] = T[0 \mathinner{.\,.} n - 1, k \mathinner{.\,.} k + m - 1]$. For every row $P[0], \ldots, P[m - 1]$ of P and every row

$T_k[0], \ldots, T_k[n-1]$ of every T_k we assign an integer identifier so that $\mathrm{Id}(P[i]) = \mathrm{Id}(T_k[j]) \Leftrightarrow P[i] = T_k[j]$ by using the KMR algorithm ([reference]) in $\tilde{\mathcal{O}}(n^2)$.

We use the approach described in [kangaroo reference]. There exists a data structure (suffix array) which for a given one-dimensional array S allows us to detect all mismatches between any given two of its subarrays of equal length. It can be built in $\tilde{\mathcal{O}}(|S|)$ and the query time is $\tilde{\mathcal{O}}(d+1)$ where d is the number of mismatches. We construct the suffix array for the concatenation of the following arrays:

- the rows $P[i]$ for every i,
- the rows $T[i]$ for every i,
- the array $\mathrm{Id}(P[0])\,\mathrm{Id}(P[1]) \ldots \mathrm{Id}(P[m-1])$,
- the arrays $\mathrm{Id}(T_k[0])\,\mathrm{Id}(T_k[1]) \ldots \mathrm{Id}(T_k[n-1])$ for every k,

the total length of which is $\mathcal{O}(n^2)$. Let us consider a problem of detecting mismatches between P and some $T' = T[j\,..\,j+m-1, k\,..\,k+m-1]$. We can first find all row indices i for which $P[i] \neq T'[i]$ by finding all mismatches between $\mathrm{Id}(P[0]) \ldots \mathrm{Id}(P[m-1])$ and $\mathrm{Id}(T_k[j]) \ldots \mathrm{Id}(T_k[j+m-1])$, which we do with query to the data structure. For every such i we can then find all mismatches between $P[i]$ and $T'[i]$ by querying $P[i]$ and $T[i+j][k\,..\,k+m-1]$. If the distance between P and T' is d, the first query takes $\tilde{\mathcal{O}}(d+1)$ operations and all subsequent queries take $\tilde{\mathcal{O}}(d+1)$ operations in total. □

Lemma 1. HD1D *with don't care symbols can be solved in* $\tilde{\mathcal{O}}(n|\Sigma|)$ *by running* $|\Sigma|$ *instances of FFT.*

Lemma 2. *There exists a* $(1+\varepsilon)$*-approximate algorithm (introduced in [3]) which solves* HD1D *with don't care symbols in* $\tilde{\mathcal{O}}(n)$*.*

Theorem 4. HD2D *with don't care symbols can be solved in* $\tilde{\mathcal{O}}(n^2|\Sigma|)$*.*

Proof. We will again use the array notation. We construct one-dimensional strings $\bar{T}$ and $\bar{P}$ by concatenating subsequent rows $T[0], \ldots, T[n-1]$ of T and rows $P[0], \ldots, P[m-1]$ of P padded with don't care symbols:

$$\bar{T} = T[0]\ T[1]\ \ldots\ T[n-1],$$

$$\bar{P} = P[0]\ ?^{n-m}\ P[1]\ ?^{n-m}\ \ldots\ ?^{n-m}\ P[m-1].$$

We run the algorithm from Lemma 1. The distance between $T[i\,..\,i+m-1, j\,..\,j+m-1]$ and P is equal to the distance between $\bar{T}[in+j\,..\,in+j+nm-n+m-1]$ and $\bar{P}$. □

Theorem 5. *There exists a* $(1+\varepsilon)$*-approximate algorithm which solves* HD2D *with don't care symbols in* $\tilde{\mathcal{O}}(n^2)$*.*

Proof. Identical to Theorem 4, but we use the algorithm from Lemma 2 instead of Lemma 1. □

The same reduction as in Theorem 4 can be applied for every HD1D solution which allows don't care symbols. Unfortunately, the most effective known algorithms for bounded HD1D rely on periodicity ([1], [2]) and inherently do not allow don't care symbols, thus, they cannot be easily generalized.

Observation 1 (Don't care padding)**.** *Every* HD2D *solution which allows don't care symbols (eg. the algorithms from Theorem 4 and Theorem 5) can be extended to also calculate the Hamming distance for occurrences of* P *which are not entirely contained in* T*. It can be done by padding the text with don't care symbols and it does not change the complexity of the solution.*

4 Proof of Theorem Theorem 1

We show an algorithm which works in time $\tilde{\mathcal{O}}(m^2 + mk^{5/4})$ assuming $2|n$ and $m < n \le \frac{3}{2}m$. By Theorem 2, our main result follows.

We start by running the algorithm from Theorem 5 with $\varepsilon = 1$. We construct the set Q as the set of such vectors $q \in \mathbb{Z}^2$ for which the estimated value of $\mathrm{Ham}(P+q,T)$ is at most $2k$. For every $q \in \{0,\dots,n-m\}^2 \setminus Q$ we say that $\mathrm{Ham}(P+q,T)$ equals ∞. The next step is to calculate the exact value of $\mathrm{Ham}(P+q,T)$ for every $q \in Q$.

Let us consider the case when $|Q| \le 2m + m^2/k$. We can run the algorithm from Theorem 3 and by the fact that $\mathrm{Ham}(P+q,T) \le 4k$ for every $q \in Q$, it will perform $\tilde{\mathcal{O}}(m^2+mk)$ operations. We are left with the case when $|Q| > 2m + m^2/k$, in which we take advantage of the fact that some strings $P+q$ for $q \in Q$ must have a large overlap and small Hamming distance from each other, and thus P must be periodic.

4.1 Two-dimensional periodicity

In this section we introduce a range of new tools related to two-dimensional periodicity. We then select some special periods of the pattern and show how to decompose it into some regularly structured monochromatic strings.

Definition 7 (Periodicity)**.** Consider any vector $\delta \in \mathbb{Z}^2$. We say that a string S has an ℓ-period δ when

$$\mathrm{Ham}(S+\delta, S) \le \ell.$$

Lemma 3. *For every $u, v \in Q$, the vector $u - v$ is an $8k$-period of P.*

Proof. $\mathrm{Ham}(P+u-v,P) = \mathrm{Ham}(P+u,P+v) \le \mathrm{Ham}(P+u,T) + \mathrm{Ham}(P+v,T) \le 4k+4k$. □

Theorem 6. *For a given $\ell \in \mathbb{Z}^+$ and a set of points $U \subseteq [\ell+1]^2$ such that $|U| > 4\ell$ there exist $s, t, s', t' \in U$ such that the following conditions hold for $w = t - s$ and $w' = t' - s'$:*

- $0 < |w||w'| \le 22\frac{\ell^2}{|U|}$,
- $|\sin\alpha| \ge \frac{1}{2}$ *where α is the angle between w and w',*
- $w, w', -w, -w'$ *are all contained in different quadrants.*

Such w, w' can be found in $\tilde{\mathcal{O}}(|U|)$ operations.

We run the algorithm from Theorem 6 on the set Q (where $\ell = n - m \le m/2$, thus $|Q| > 2m + m^2/k \ge 4\ell$). We obtain vectors $\varphi \in \mathcal{Q}_4$ and $\psi \in \mathcal{Q}_1$ which by Lemma 3 are $\mathcal{O}(k)$-periods of P. We will refer to those vectors throughout the rest of the description and we define $p = \varphi \times \psi$. Note that because $|Q| > 2m + m^2/k$, we have $p \le |\varphi||\psi| = \mathcal{O}(\min\{m,k\})$.

Definition 8 (Lattice congruency)**.** We define $\mathcal{L} = \{s\varphi + t\psi : s,t \in \mathbb{Z}\}$. We say that two vectors $u, v \in \mathbb{Z}^2$ are **lattice-congruent** and denote $u \equiv v$ when $u - v \in \mathcal{L}$ [Galil citation].

Lemma 4. *There exists a set of points $\Gamma \subseteq \mathbb{Z}^2$ such that $|\Gamma| = p$ and every point $u \in \mathbb{Z}^2$ is lattice-congruent to exactly one point $\gamma \in \Gamma$.*

Definition 9 (Parquet)**.** We call a non-empty set $U \subseteq \mathbb{Z}^2$ a **parquet** when there exist some values $x_0, x_1, y_0, y_1, \varphi_0, \varphi_1, \psi_0, \psi_1 \in \mathbb{Z}$, which we will call its **signature**, such that

$$U = [x_0, x_1) \times [y_0, y_1) \cap \{u : u \in \mathbb{Z}^2, \varphi \times u \in [\varphi_0, \varphi_1), \psi \times u \in [\psi_0, \psi_1)\}.$$

a) If additionally $x_1 - x_0 \ge |\varphi.x| + |\psi.x|$ and $y_1 - y_0 \ge |\varphi.y| + |\psi.y|$, then U is a **spacious** parquet.

b) If additionally $x_0, y_0 = -\infty$ and $x_1, y_1 = +\infty$, then U is a **simple** parquet.

Note that every simple parquet is spacious.

Definition 10 (Subparquet)**.** We call a non-empty set $V \subseteq \mathbb{Z}^2$ a **subparquet** when there exists a parquet U and a point $\gamma \in \mathbb{Z}^2$ such that

$$V = \{\, u : u \in U, u \equiv \gamma \,\}.$$

We call V a spacious/simple subparquet when there exists U which is (correspondingly) a spacious/simple parquet. We say that V is lattice-congruent to some $v \in \mathbb{Z}^2$ (denoted as $V \equiv v$) when $v \equiv \gamma$. We similarly define lattice congruency between two subparquets.

Definition 11 (Parquet string)**.** We call a string S a spacious/simple (sub-)parquet string when $\operatorname{dom}(S)$ is a spacious/simple (sub-)parquet.

Theorem 7 (Periodic string decomposition)**.** *A given spacious/simple parquet string R with $\mathcal{O}(k)$-periods φ and ψ can be partitioned in time $\tilde{\mathcal{O}}(|\operatorname{dom}(R)|)$ into $\mathcal{O}(k)$ monochromatic spacious/simple subparquet strings, correspondingly.*

Since $|\varphi.x|, |\varphi.y|, |\psi.x|, |\psi.y| \leq n - m \leq m/2$, the $m \times m$ string P is a spacious parquet string and satisfies the assumptions of Theorem 7. We partition P into a set of monochromatic spacious subparquet strings $\mathcal{V}$, where $|\mathcal{V}| = \mathcal{O}(k)$. Note that because the text is not necessarily periodic, we unfortunately cannot use the same approach for T.

4.2 Text decomposition

In this section we show how to decompose the text using a similar but more nuanced approach. We then introduce an effective way to aggregate the contributions of every pair of strings that P and T are decomposed into.

Definition 12 (Active text)**.** Let $A = \bigcup_{q \in Q} \operatorname{dom}(P) + q$ and $B = \operatorname{dom}(T) \setminus A$. We define the **active text** as a string $T_{\mathbf{a}} = T\restriction_A$ and the **inactive text** as a string $T_{\mathbf{b}} = T\restriction_B$. For a point $u \in \mathbb{Z}^2$ we define its **border distance** as $\min\{\, \|u - v\|_\infty : v \in B \,\}$, which we will denote as $\operatorname{BD}(u)$. For a set of points $U \subseteq \mathbb{Z}^2$ we define $\operatorname{BD}(U) = \max\{\, \operatorname{BD}(u) : u \in U \,\}$. Note that we consider the maximum distance, not minimum.

Observation 2. $\operatorname{Ham}(P + q, T) = \operatorname{Ham}(P + q, T_{\mathbf{a}})$ *for every* $q \in Q$.

Theorem 8 (Active text decomposition)**.** *For a given parameter $\ell \in ?$ the active text can be partitioned in time $\tilde{\mathcal{O}}(m^2)$ into a set of $\mathcal{O}(\min\{\, m^2, \ell k \,\})$ monochromatic simple subparquet strings and a string F such that $\operatorname{BD}(F) = \mathcal{O}(m/\ell)$.*

Theorem 9. *For a given list of signatures of simple subparquets $U_1, \ldots, U_\ell$, list of signatures of subparquets $V_1, \ldots, V_\ell$ and a set of vectors Q we can calculate*

$$\sum_{i=1}^{\ell} |(U_i + q) \cap V_i|$$

for every $q \in Q$ in total time $\tilde{\mathcal{O}}(n^2 + \ell + |Q|)$ assuming that the subparquets only contain vectors of length $\mathcal{O}(n)$.

Theorem 10. *For a given string F such that* $\operatorname{dom}(F) \subseteq \operatorname{dom}(T_{\mathbf{a}})$ *we can calculate* $\operatorname{Ham}(P + q, F)$ *for every $q \in Q$ in total time $\tilde{\mathcal{O}}(m^2 + mk^{1/2}\operatorname{BD}(F))$.*

We partition $T_{\mathbf{a}}$ using the algorithm from Theorem 8 with $\ell = mk^{-3/4}$ into a set of simple subparquet strings $\mathcal{U}$ and a string F. For every $q \in Q$ we then have

$$\operatorname{Ham}(P + q, T_{\mathbf{a}}) = \operatorname{Ham}(P + q, F) + \sum_{U \in \mathcal{U}} \sum_{V \in \mathcal{V}} \operatorname{Ham}(U - q, V)$$

4.3 Proof of Theorem 6

First, we find any closest pair of vectors $s, t \in U$ by running the standard $\tilde{\mathcal{O}}(|U|)$ time algorithm and denote $w = t - s$. We define a partial order $\leq_w$ where $v \leq_w u$ for some $u, v \in U$ when at least one condition holds:

(a) $u = v$,

(b) $u - v$ and w belong to the same quadrant,

(c) $\alpha \in (-\frac{\pi}{6}, \frac{\pi}{6})$ where α is the angle between w and $u - v$.

We find the longest chain C and the longest antichain A using dynamic programming in $\tilde{\mathcal{O}}(|U|)$ operations. We then find any closest pair of vectors $s', t' \in A$ and denote $w' = t' - s'$. We have the following inequalities:

(i) $|U| \leq |C||A|$ (by Dilworth's theorem),

(ii) $(|C| - 1)|w| \leq (1 + \sqrt{3})\ell$ (roughly by the fact that vectors in C must be increasing in a certain direction),

(iii) $(|A| - 1)|w'| \leq 2\ell$ (by using a similar argument for vectors in A).

By considering the assumption $|U| > 4\ell$ it can be proven that $|w||w'| \leq 22\frac{\ell^2}{|U|}$ and the other conditions also hold.

4.4 Proof of Theorem 7

Definition 13 (Lattice graph)**.** For a set $U \subseteq \mathbb{Z}^2$ we define its **lattice graph** $G_U = (U, E_U)$ where

$$E_U = \left\{ \{ u, u + \delta \} : \delta \in \{ \varphi, \psi \}, u \in U, u + \delta \in U \right\}$$

so every vector is connected with its translations by $\varphi, \psi, -\varphi, -\psi$.

Lemma 5. *If U is a spacious subparquet, then G_U is connected.*

Firstly, we partition R into a set of subparquet strings $\mathcal{S}$. For every non-empty $S \in \mathcal{S}$ we consider a lattice graph $G_{\mathrm{dom}(S)}$. If S is not monochromatic, then since $G_{\mathrm{dom}(S)}$ is connected, there must exist a pair of neighboring vectors v, w such that $S(v) \neq S(w)$. We select any such pair and partition S into spacious (or simple if S is simple) subparquet strings S' and S'' such that $v \in S'$ and $w \in S''$. For example if $v = w + \varphi$, then $S' = \{ u : u \in S, \psi \times u \leq \psi \times v \}$ and $S'' = \{ u : u \in S, \psi \times u > \psi \times v \}$. In the cases when $v = w + \delta$ for $\delta \in \{ -\varphi, \psi, -\psi \}$ the construction in similar.

We can recursively partition S' and S'' further until we obtain monochromatic strings. Because R has $\mathcal{O}(k)$-periods φ and ψ, the total number of neighbor pairs v, w such that $S(v) \neq S(w)$ is $\mathcal{O}(k)$ throughout all $S \in \mathcal{S}$. Thus the total number of recursive calls is $\mathcal{O}(k)$ and because $|\mathcal{S}| = \mathcal{O}(k)$, the total number of constructed strings is $\mathcal{O}(k)$. The algorithm can be implemented to work in time $\tilde{\mathcal{O}}(|\operatorname{dom}(R)|)$.

4.5 Proof of Theorem 8

We assume ℓ to be an even number smaller than $\frac{n}{4}$ (if it is not, we can find $\ell' = \Theta(\ell)$, which is). We start by partitioning $\mathrm{dom}(T)$ into **tiles**. We define $\varphi_{\min} = \min \{ \varphi \times u : u \in \mathrm{dom}(T) \}$, analogously $\varphi_{\max}, \psi_{\min}, \psi_{\max}$ and denote $\delta_\varphi = \frac{\varphi_{\max} - \varphi_{\min}}{\ell}$, $\delta_\psi = \frac{\psi_{\max} - \psi_{\min}}{\ell}$. We define a tile with integer coordinates (s, t) as a set of vectors $u \in \mathbb{Z}^2$ such that

$$\varphi_{\min} + s\delta_\varphi < \varphi \times u \leq \varphi_{\min} + (s + 1)\delta_\varphi,$$

$$\psi_{\min} + t\delta_\psi < \psi \times u \le \psi_{\min} + (t+1)\delta_\psi.$$

For a fixed tile U consider $x_{\min} = \min\{\, u.x : u \in U \,\}$, analogously $x_{\max}, y_{\min}, y_{\max}$ and a set

$$R = \{\, u : u \in \mathbb{Z}^2, x_{\min} \le u.x \le x_{\max}, y_{\min} \le u.y \le y_{\max} \,\}.$$

We classify U into one of three types:

a) if $U \cap T_{\mathbf{a}} = \emptyset$ then U is an inactive tile,

b) if $U \cap T_{\mathbf{a}} \neq \emptyset$, $R \nsubseteq T_{\mathbf{a}}$ then U a border tile,

c) if $U \cap T_{\mathbf{a}} \neq \emptyset$, $R \subseteq T_{\mathbf{a}}$ then U is an active tile.

We define B as a set of all $u \in T_{\mathbf{a}}$ contained in a border tile and construct $F = (B, T^{\mathbf{f}})$. Let us denote $z = \frac{n-1}{2}$. Consider a family of sets $\mathcal{R} = \{\, R_i^1 \,\} \cup \{\, R_i^2 \,\} \cup \{\, R_i^3 \,\} \cup \{\, R_i^4 \,\}$, where for every active tile U with coordinates (s,t) its members are placed into exactly one subset:

1) R_t^1 if $y_{\min} > z$, $x_{\max} \ge z$,

2) R_s^2 if $x_{\max} < z$, $y_{\max} \ge z$,

3) R_t^3 if $y_{\max} < z$, $x_{\min} \le z$,

4) R_s^4 if $x_{\min} > z$, $y_{\min} \le z$.

The number of non-empty sets $R \in \mathcal{R}$ is $\mathcal{O}(\ell)$. For each of them we consider $S = (R, T^{\mathbf{f}})$ which is a simple parquet string with $\mathcal{O}(k)$-periods φ and ψ, and we further partition it using algorithm from Theorem 7, thus constructing the set $\mathcal{U}$.

4.6 Proof of Theorem 9

Throughout this section we will denote $D = \{\, u : u \in \mathcal{L}, \varphi \times u \ge 0, \psi \times u \ge 0 \,\}$, where $\mathcal{L}$ is the set introduced in Definition 8.

Lemma 6. *Given a set of subparquets $\mathcal{V}$ and a set of points Q, we can calculate*

$$\sum_{V \in \mathcal{V}} |(D+q) \cap V|$$

for every $q \in Q$ in total time $\tilde{\mathcal{O}}(n^2 + |Q| + |\mathcal{V}|)$ assuming that every $V \in \mathcal{V}$ consists of vectors of length $\mathcal{O}(n)$.

Proof. For every $u \in \mathbb{Z}^2$ let us define $\text{score}(u) = |\{\, V : V \in \mathcal{V}, u \in V \,\}|$. Observe that

$$\sum_{V \in \mathcal{V}} |(D+q) \cap V| = \sum_{u \in D+q} \text{score}(u).$$

We start by explicitly calculating the scores. We find the maximum length of a vector that some $V \in \mathcal{V}$ is defined for, which we denote ℓ. We construct the set $U \subseteq \mathbb{Z}^2$ of all vectors of length at most ℓ. By the assumption, we have $\ell = \mathcal{O}(n)$, and thus $|U| = \mathcal{O}(l^2) = \mathcal{O}(n^2)$. We observe that since all the scores are zero for points outside of U, we can only calculate them for $u \in U$.

We find the set Γ introduced in Lemma 4 and for every $\gamma \in \Gamma$ we construct $U_\gamma = U \cap (\mathcal{L} + \gamma)$. Consider any $u \in U_\gamma$ for some fixed $\gamma \in \Gamma$ and any $V \in \mathcal{V}$. We observe that if $V \not\equiv \gamma$, then $u \notin V$ and thus V does not contribute to $\text{score}(u)$. If $V \equiv \gamma$, then we can find a parquet W such that $V = W \cap (\mathcal{L} + \gamma)$ and we have $u \in V \Leftrightarrow u \in W \cap (\mathcal{L} + \gamma) \Leftrightarrow u \in W$. Thus, if we denote $\mathcal{W}_\gamma$ as the set of parquets W obtained for every $V \in \mathcal{V}$ such that $V \equiv \gamma$, then $\text{score}(u)$ for $u \in U_\gamma$ is the number of parquets $W \in \mathcal{W}_\gamma$ such that $u \in W$. We calculate $\text{score}(u)$ for every $u \in U_\gamma$ by sweeping U_γ and $\mathcal{W}_\gamma$ in time $\tilde{\mathcal{O}}(|U_\gamma| + |\mathcal{W}_\gamma|)$. We do it independently for every $\gamma \in \Gamma$, performing $\tilde{\mathcal{O}}(|U| + |\mathcal{V}|) = \tilde{\mathcal{O}}(n^2 + |\mathcal{V}|)$ operations in total.

Now consider a query vector $q \in Q$. Let $\gamma \in \Gamma$ be such that $q \equiv \gamma$. We have already showed that the sum of scores for $u \in D + q$ is equal to the sum of scores for $u \in (D + q) \cap U$. Since $(D + q) \cap U = (D + q) \cap U_\gamma$, we see that the result is the sum of scores for $u \in U_\gamma$ such that $\varphi \times u \geq \varphi \times q$ and $\psi \times u \geq \psi \times q$. If we denote $Q_\gamma = Q \cap (\mathcal{L} + \gamma)$, we see that we can calculate the results for all $q \in Q_\gamma$ by sweeping Q_γ and U_γ in time $\tilde{\mathcal{O}}(|Q_\gamma| + |U_\gamma|)$. We do it independently for every $\gamma \in \Gamma$, performing $\tilde{\mathcal{O}}(|Q| + |U|) = \tilde{\mathcal{O}}(n^2 + |Q|)$ operations in total. $\square$

Lemma 7. *For any simple subparquet U we can find $w_0, \ldots, w_3 \in \mathbb{Z}^2$ such that*

$$|U \cap X| = \sum_{j=0}^{3} (-1)^j |(D + w_j) \cap X|$$

for every $X \subseteq \mathbb{Z}^2$. If U consists of vectors of length $\mathcal{O}(n)$, then $w_0, \ldots, w_3$ are of length $\mathcal{O}(n)$.

We apply Lemma 7 to every U_i and find $w_{i,0}, \ldots, w_{i,3}$ so that we have

$$\begin{aligned}\sum_{i=1}^{\ell} |(U_i + q) \cap V_i| = \sum_{i=1}^{\ell} |U_i \cap (V_i - q)| &= \sum_{i=1}^{\ell} \sum_{j=0}^{3} (-1)^j |(D + w_{i,j}) \cap (V_i - q)| \\ &= \sum_{j=0}^{3} (-1)^j \sum_{i=1}^{\ell} |(D + q) \cap (V_i - w_{i,j})|.\end{aligned}$$

By Lemma 6 we can independently calculate the values $\sum_{i=1}^{\ell} |(D + q) \cap (V_i - w_{i,j})|$ for every j by running the algorithm for $\mathcal{V}_j = \{ V_i - w_{i,j} : i \in \{ 1, \ldots, \ell \} \}$ and Q.

4.7 Proof of Theorem 10

Recall that by Theorem 7 we can partition P into a set of monochromatic subparquet strings $\mathcal{V}$, where $|\mathcal{V}| = O(k)$. For every character $\sigma \in \Sigma$ present in P we construct $\mathcal{V}_\sigma = \{ V : V \in \mathcal{V}, \mathrm{C}(V) = \sigma \}$. We call σ a **frequent** character if $|\mathcal{V}_\sigma| \geq \sqrt{k}$ and if $|\mathcal{V}_\sigma| < \sqrt{k}$, we call it an **infrequent** character. We partition F into two strings S and R, where $S(u)$ is a frequent character for every $u \in S$ and $R(u)$ is an infrequent character for every $u \in R$.

For every $q \in Q$ we then have $\mathrm{Ham}(P + q, F) = \mathrm{Ham}(P + q, S) + \mathrm{Ham}(P + q, R)$, which we calculate independently.

4.7.1 Frequent character contribution

We show how to calculate $\mathrm{Ham}(P + q, S)$ for every $q \in Q$. We start by partitioning S into four strings $S_1, \ldots, S_4$ by splitting through the middle with a horizontal and vertical line. Specifically

- S_1 is the restriction of S to $\{ 0, \ldots, n/2 - 1 \} \times \{ 0, \ldots, n/2 - 1 \}$,
- S_2 is the restriction of S to $\{ 0, \ldots, n/2 - 1 \} \times \{ n/2, \ldots, n - 1 \}$,
- S_3 is the restriction of S to $\{ n/2, \ldots, n - 1 \} \times \{ n/2, \ldots, n - 1 \}$,
- S_4 is the restriction of S to $\{ n/2, \ldots, n - 1 \} \times \{ 0, \ldots, n/2 - 1 \}$.

We independently calculate $\mathrm{Ham}(P + q, S_i)$ for every i and sum the results. We show how to calculate $\mathrm{Ham}(P + q, S_1)$ for every $q \in Q$. The other components can be handled similarly.

We will take advantage of the fact that the points close to the border can overlap only with a small subset of points from the pattern when considering the occurrences fully contained in the active text. Specifically, consider a set $J = [d] \times [n] \cup [n] \times [d]$ and a string $P_J = P \restriction_J$.

Lemma 8. $\mathrm{Ham}(P+q,S_1) = \mathrm{Ham}(P_J+q,S_1)$ *for every* $q \in Q$.

Theorem 11. *We can partition* S_1 *into* $\mathcal{O}(m/d)$ *strings such that their width is* $\mathcal{O}(d)$ *and the sum of their heights is* $\mathcal{O}(m)$.

4.7.2 Proof of Theorem 11

Consider an array of strings $U_0, \ldots, U_{\lceil n/d \rceil - 1}$ where U_i is the restriction of S_1 to $\{ id, \ldots, id+d-1 \} \times [n] \cap \mathrm{dom}(S_1)$. For the sake of formality (since the maximum/minimum of an empty set is undefined) we construct an array of strings $V_0, \ldots, V_{\ell-1}$ consisting of all non-empty strings U_i, given in the ascending order of i. Observe that $V_0, \ldots, V_{\ell-1}$ partition S_1, their width is $\mathcal{O}(d)$ and it remains to show that the sum of their heights is $\mathcal{O}(m)$. Let us denote $a_i = \min\{ u.y : u \in V_i \}$ and $b_i = \max\{ u.y : u \in V_i \}$. Our goal is to prove that $\sum_{i=0}^{\ell-1} b_i - a_i + 1 = \mathcal{O}(m)$.

Lemma 9. $b_{i+2} - a_i \leq d$ *for every* $i < \ell - 2$.

Proof. Let us assume the contrary and have $b_{i+2} - a_i > d$ for some i. There exists $u \in V_i$ such that $u.y = a_i$ and $v \in V_{i+2}$ such that $v.y = b_{i+2}$. We have $v.y - u.y > d$ (by assumption) and $v.x - u.x > d$ (since $u \in V_i$, $v \in V_{i+2}$). □

4.7.3 Infrequent character contribution

References

[1] Raphaël Clifford, Allyx Fontaine, Ely Porat, Benjamin Sach, and Tatiana Starikovskaya. The k-mismatch problem revisited. *CoRR*, abs/1508.00731, 2015.

[2] Pawel Gawrychowski and Przemyslaw Uznanski. Optimal trade-offs for pattern matching with k mismatches. *CoRR*, abs/1704.01311, 2017.

[3] Howard J. Karloff. Fast algorithms for approximately counting mismatches. *Inf. Process. Lett.*, 48(2):53–60, 1993.