Hardness of Optimal Spaced Seed Design

François Nicolas and Eric Rivals

L.I.R.M.M.
University of Montpellier II, CNRS U.M.R. 5506
161 rue Ada, F-34392 Montpellier Cedex 5, France
{nicolas,rivals}@lirmm.fr

Abstract. Speeding up approximate pattern matching is a line of research in stringology since the 80's. Practically fast approaches belong to the class of filtration algorithms, in which text regions dissimilar to the pattern are excluded (filtered out) in a first step, and remaining regions are compared to the pattern by dynamic programming in a second step. Among the necessary conditions used to test similarity between the regions and the pattern, many require a minimum number of common substrings between them. When only substitutions are taken into account for measuring dissimilarity, it was shown recently that counting spaced subwords instead of substrings improve the filtration efficiency. However, a preprocessing step is required to design one or more patterns, called gapped seeds, for the subwords, depending on the search parameters. The seed design problems proposed up to now differ by the way the similarities to detect are given: either a set of similarities is given *in extenso* (this is a "region specific" problem), or one wishes to detect all similar regions having at most k substitutions (general detection problem). Several articles exhibit exponential algorithms for these problems. In this work, we provide hardness and inapproximability results for both the region specific and general seed design problems, thereby justifying the exponential complexity of known algorithms. Moreover, we introduce a new formulation of the region specific seed design problem, in which the weight of the seed (*i.e.*, number of characters in the subwords) has to be maximized, and show it is as difficult to approximate than Maximum Independent Set.

1 Introduction

A routine task in computational genomics is to search among all known sequences those being similar to a sequence of interest. "Similar" means that can be aligned over reasonably long portions. The similarity in sequence helps in the annotation of the sequence of interest as it may reveal, *e.g.*, if it is a gene, a similarity in function, in regulation, in its interaction with other molecules, in three dimensional structure of the protein product, or a common origin. This task is known as sequence similarity search. Since the 90's, heuristic algorithms [1] are preferred to the direct application of dynamic programming schemes, which require quadratic time. In practice, as the size of the sequence databases

A. Apostolico, M. Crochemore, and K. Park (Eds.): CPM 2005, LNCS 3537, pp. 144–155, 2005.

grows exponentially, efficiency is achieved by *filtration*. The underlying principle of filtration is to exclude in a first step regions of the sequence database that are surely not similar to the query sequence by testing a simple necessary condition. The second step performs an alignment procedure by dynamic programming with the few remaining regions. Application of filtration occurs in software like FLASH [5] or QUASAR [3].

Usual necessary conditions rely on counting common contiguous subwords to the query and the database sequence. Recently, several authors research have emphasized that the shape of the subwords plays a key role in filtration efficiency, and proposed to use "carefully chosen arbitrary shapes" for the subwords [4, 11–14]. The shape of the subwords is given by a gapped seed, *e.g.*, a pattern like `##-##--#` where the `#` symbol indicates which position should match between the query's and database sequence's subword, and the `-` are don't care positions. One central problem is to choose such a seed to optimize the filtration efficiency. Given a set of similarities (alignments) of interest, the goal is to find the best seed or family of seeds. The problem has been declined in several formulations, either as a decision or a maximization problem. In the former, one searches for a seed that detects all similarities, in the latter for a seed that detects all similarities and maximizes, *e.g.*, the number of `#` (its weight). Several algorithms whose complexity depends exponentially on the length of the seed have been proposed to solve these problems, but it is not known to which complexity class the simplest forms of the seed design problem belong. Our article answer these questions by showing these problems are NP-hard, or even worse, difficult to approximate.

In [12], the case of lossy filtration is investigated. The authors show that computing the hit probability of set of seeds is NP-hard, but can be approximated (admits a PTAS). They also prove the NP-hardness and the inapproximability of a region specific multiple seeds design problem, where both the set of similarities and the weight of the seeds are constrained (see problem RSOS below).

In this abstract, we consider both lossy and lossless filtrations. We improve on the results of [12] by showing the inapproximability of RSOS even in the case of a single seed (Section 4). Moreover, we prove the hardness of a general seed design problem: NON DETECTION as defined in [11] (Section 2). In this problem, one considers the set of all similarities at a given Hamming distance from the query (this is independent on the query). The problem is more general than RSOS. As by-product of our proof, we introduce and classify a tiling problem (SSC). Several works [4, 7, 11] give empirical and theoretical evidences that support the correlation between the weight of the seed and filtration efficiency. Building on this idea, we propose an optimization problem MWLS in which the weight of the designed seed has to be maximized. We provide a proof of NP-hardness and of inapproximability for MWLS (Section 3).

In the remaining of this section, we introduce a notation, define the investigated problems, and survey known results. Sections 2, 3, and 4 are each dedicated to a problem as listed above and are independent of each other.

1.1 Definitions and Problems

Let $\mathbb{Z}$ denote the set of all integers and for any a, $b \in \mathbb{Z}$, let $[a, b]$ be the set of all $x \in \mathbb{Z}$ satisfying $a \leq x \leq b$. For any finite set X, we denote by $\#X$ the *cardinality* of X. An *alphabet* Σ is a finite set of *letters*. A *word* or *string* over Σ is a finite sequence of elements of Σ. The set of all words over Σ is denoted by $\Sigma^\star$. For a word x, $|x|$ denotes the *length* of x. Given two words x and y, we denote by xy the *concatenation* of x and y. For every $1 \leq i \leq j \leq |x|$, $x[i]$ denotes the i-th letter of x, and $x[i\ ;\ j]$ denotes the *substring* $x[i]x[i+1] \ldots x[j]$. For every letter a, $|x|_a := \#\{i \in [1, |x|] : x[i] = a\}$ denotes the number of *occurrences* of the letter a in x. For every integer $n \geq 0$, x^n denotes the concatenation of n copies of x.

Definition 1 (Weight, seed). *The* weight *of a word* $w \in \{\texttt{\#}, \texttt{-}\}^\star$*, denoted* $\|w\|$*, is the number of occurrences of the symbol* # *in* w*. A* seed *is a non empty word over the alphabet* $\{\texttt{\#}, \texttt{-}\}$ *and whose first and last letter is a* # *(*i.e.*, an element of* $\texttt{\#}\{\texttt{\#}, \texttt{-}\}^\star\texttt{\#} \cup \{\texttt{\#}\}$*).*

Definition 2 (Similarity). *A* similarity *is a word over* $\{\texttt{0}, \texttt{1}\}$*. Let* m*,* k *be two integers such that* $0 \leq k \leq m$*. An* (m, k)-similarity *is a similarity of length* m *with* k *occurrences of the symbol* 0 *and* $m - k$ *occurrences of the symbol* 1 *(*i.e.*, an element of* $\{s \in \{\texttt{0}, \texttt{1}\}^m : |s|_{\texttt{0}} = k\}$*).*

Definition 3 (Detection). *Let* g *be a word over* $\{\texttt{\#}, \texttt{-}\}$*,* Γ *a set of words over* $\{\texttt{\#}, \texttt{-}\}$*, and* s *a similarity. Let* $i \in [0, |s| - |g|]$*. We say that* g detects s at position i *if, for all* $j \in [1, |g|]$*,* $s[i + j] = \texttt{1}$ *whenever* $g[j] = \texttt{\#}$*. We say that* g detects s *whenever there exists* $i \in [0, |s| - |g|]$ *such that* g *detects* s *at position* i*. Moreover,* Γ detects s *if there is* $g \in \Gamma$ *that detects* s*.*

Note that in the previous definition, g (resp. Γ) may be a seed (resp. a set of seeds). In the sequel, ϵ denotes an arbitrarily small positive real number.

We study the complexity of three problems. In Section 2, we consider the decision problem [11]:

Name: Non Detection
Instance: A seed g, two integers m and k satisfying $0 \leq k \leq m$.
Question: Does it exist an (m, k)-similarity not detected by g?

We show that Non Detection is NP-complete (Theorem 2). Note that we assume that any instance (g, m, k) has size $O(|g| + m)$, *i.e.*, that the integers m and k are encoded in unary. If encoded in binary (as usual), (g, m, k) would have size only $O(|g| + \log m)$. In other words, we demonstrate that Non Detection is *strongly* NP-complete.

In Section 3, we investigate the difficulty to approximate the maximization problem:

Name: region specific Maximum Weight Lossless Seed (MWLS)
Instance: A finite set S of similarities.

Solution: A seed g that detects all similarities of S.

Measure: The weight of g.

Theorem 3 proves it does not exist a polynomial time approximation algorithm for MWLS with bound $(\#S)^{0.25-\epsilon}$ unless P = NP.

In Section 4, we study the maximization problem [12]:

Name: Region Specific Optimal Seeds (RSOS)

Instance: Two integers d and p, a finite set S of similarities.

Solution: A set Γ of seeds satisfying $\#\Gamma = d$ and $\|g\| = p$ for any $g \in \Gamma$.

Measure: The number of similarities in S detected by Γ.

Theorem 4 states that, even when restricted to instances (d, p, S) such that $d = 1$, it does not exist a polynomial time approximation algorithm for RSOS with bound $\frac{e}{e-1} - \epsilon$.

We need additional definitions on hypergraphs. A *hypergraph* is a pair $H := (V, \mathcal{E})$ where V is a finite set of *vertices* and $\mathcal{E}$ is a set of subsets of V. The elements of $\mathcal{E}$ are called *hyperedges*. An *independent set* I of H is a subset of V such that for any $E \in \mathcal{E}$, one has $E \not\subseteq I$. Let $r \geq 2$ be an integer. H is said to be r-uniform when for any $E \in \mathcal{E}$, $\#E = r$. A 2-uniform hypergraph is a *graph* and its hyperedges are simply called *edges*.

1.2 Related Works

Concerning Non Detection**.** Let m and k be two integers such that $0 \leq k \leq m$. Let us denote by

- $\mathrm{U}(\Gamma, m, k)$ the number of (m, k)-similarities left undetected by the set of seeds Γ,
- $\mathrm{T}(\Gamma, m, k)$ the largest integer $t \geq 0$ satisfying: for any (m, k)-similarity s, there are t distinct pairs (i_1, g_1), (i_2, g_2), …, (i_t, g_t) such that for any $j \in [1, t]$ one has $g_j \in \Gamma$, $i_j \in [0, |s| - |g_j|]$ and g_j detects s at position i_j. Informally, $\mathrm{T}(\Gamma, m, k)$ is the minimal number of positions at which any (m, k)-similarity is detected by Γ.

In [11], one finds dynamic programming algorithms to compute $\mathrm{U}(\Gamma, m, k)$ and $\mathrm{T}(\Gamma, m, k)$ in time proportional to

$$m \times \sum_{j=0}^{k} \binom{\lambda}{j} (k - j + 1) + (\#\Gamma) \times \sum_{j=0}^{k} \binom{\lambda}{j} \quad \text{with} \quad \lambda := \max_{g \in \Gamma} |g| \ .$$

A simple bound [4] guarantees that these algorithms have complexities in $O\big(2^{\lambda} \times (mk + \#\Gamma)\big)$, and are thus Fixed Parameter Tractable (FPT) for parameter λ (see [6] for details on parameterized complexity). The algorithm that computes $\mathrm{T}(\Gamma, m, k)$ described in [11] generalizes to a family of seeds the one for the case of a single seed given in [4, Section 4].

Solving Non Detection for an instance (g, m, k) means to decide whether $\mathrm{U}(\{g\}, m, k)$ differs from zero (resp. if $\mathrm{T}(\{g\}, m, k)$ equals zero). Theorem 2 implies that, even if we restrict ourselves to the case of a single seed ($\#\Gamma = 1$), any algorithm computing $\mathrm{U}(\Gamma, m, k)$ (resp. $\mathrm{T}(\Gamma, m, k)$) requires in the worst case exponential time. Thus, the algorithms given in [4, 11] have the best time complexities one can hope.

Concerning MWLS **and** RSOS**.** It is shown in [12] that the decision version of RSOS is NP-hard, even when searching for a single seed instead of a family of seeds. The authors of [12] also prove that RSOS does not admit a polynomial time approximation algorithm with bound $\frac{e}{e-1} - \epsilon$ unless P = NP. Theorems 3 and 4 improve on these results.

2 Hardness of Non Detection

To show the hardness of Non Detection, we introduce an intermediate problem:

Name: Soapy Set Cover (SSC)
Instance: A finite subset $G \subseteq \mathbb{Z}$, two non-negative integers N and q.
Question: Does it exist a subset $T \subseteq \mathbb{Z}$ of cardinality q such that $G + T$ contains at least N consecutive integers?

It is related to tiling problems. We assume that any instance (G, N, q) of SSC has size $O(\max G - \min G + N + q)$. In other words, we assume that the integers N and q are encoded in unary, and that the set G is encoded by a bit-vector.

First in Theorem 1, we reduce Exact Cover by 3-Sets (X3C) to SSC. In X3C, we are given a 3-uniform hypergraph $(V, \mathcal{E})$ and search for a subset of $\mathcal{E}$ that partitions V. X3C is NP-hard [9] (it can be seen as a generalization of 3D-Matching). Then, in Theorem 2 we reduce SSC to Non Detection.

Theorem 1. SSC *is* NP*-complete.*

Proof. SSC is in NP, since for any positive instance (G, N, q) of SSC, a subset $T \subseteq [1 - \max G, N - \min G]$ of cardinality at most q satisfying $[1, N] \subseteq G + T$ is a polynomial certificate for SSC on (G, N, q). Let us now reduce X3C to SSC.

Let $(V, \mathcal{E})$ be an instance of X3C. If 3 does not divide $\#V$ then $(V, \mathcal{E})$ is a negative instance of X3C that we transform into $(\emptyset, 1, 0)$, a negative instance of SSC. Without loss of generality, we can now suppose that after numbering the elements of V, $V = [q + 1, 4q]$ where $q := \frac{\#V}{3}$. Let us also number the elements of $\mathcal{E}$: set $m := \#\mathcal{E}$ and write $\mathcal{E} = \{E_1, E_2, \ldots, E_m\}$.

Let $N := 2q^2 + 4q$. For any $i \in [1, m]$ and any $j \in [1, q]$, let:

$$F_j := [(j-1)(2q-1) + 4q + 1, j(2q-1) + 4q] ,$$
$$G_{i,j} := \{j\} \cup E_i \cup F_j \cup \{N - j + 1\} , \qquad \tau_{i,j} := 2N((i-1)q + j - 1) ,$$

and set $G := \bigcup_{i=1}^{m} \bigcup_{j=1}^{q} (G_{i,j} + \tau_{i,j})$.

We obtain an instance (G, N, q) of SSC. One can easily check that this transformation takes polynomial time.

Let us first explain the gadget of the proof. The sets $G_{i,j}$ (for $(i, j) \in [1, m] \times [1, q]$) are subsets of $[1, N]$, and the $\tau_{i,j}$'s are the mq multiples of $2N$ comprised between 0 and $2N(mq - 1)$. Thus, G is a subset of $[1, 2Nmq]$. Moreover, each of the mq intervals of length $2N$ partitioning $[1, 2Nmq]$ (that is to say, the $[2N(k-1)+1, 2Nk]$'s for $k \in [1, mq]$) contain a unique $G_{i,j} + \tau_{i,j}$ in their left half and no element of G in their right half.

Let us now dwell on the $G_{i,j}$: the cardinality of $G_{i,j}$ is $2q+4$ since $G_{i,j}$ is the disjoint union of the hyperedge E_i whose cardinal is 3, of the segment F_j whose cardinality equals $2q - 1$, and of two singletons.

Let $F := [4q+1, N-q] = \left[4q+1, 2q^2+3q\right]$. The four segments $[1, q]$, $[q+1, 4q]$, F, and $[N-q+1, N]$ have length q, $3q$, $2q^2 - q$, and q, respectively. They form a partition of $[1, N]$. Each contributes to $G_{i,j}$: the singleton $\{j\}$ is included in $[1, q]$, the hyperedge E_j is included in $[q+1, 4q] = V$, F_j is included in F, and the singleton $\{N - j + 1\}$ in $[N - q + 1, N]$. Besides, $\{F_1, F_2, \ldots, F_q\}$ is the unique partition of F in segments of length $2q - 1$.

Lemma 11. *If $(V, \mathcal{E})$ is a positive instance of* X3C *then (G, N, q) is a positive instance of* SSC*.*

Proof. Suppose there exists $\mathcal{C} \subseteq \mathcal{E}$ that is a partition of V. Then, $\mathcal{C}$ has cardinality $\#V/3 = q$ and thus, there are $i_1, i_2, \ldots, i_q \in [1, m]$ such that $\mathcal{C} = \{E_{i_1}, E_{i_2}, \ldots, E_{i_q}\}$. Let us set $T := \left\{-\tau_{i_1,1}, -\tau_{i_2,2}, \ldots, -\tau_{i_q,q}\right\}$.

By construction, T has cardinality q and for any $j \in [1, q]$, one has

$$G_{i_j,j} = (G_{i_j,j} + \tau_{i_j,j}) - \tau_{i_j,j} \subseteq G - \tau_{i_j,j} \subseteq G + T$$

therefore $G + T$ includes

$$\begin{aligned}\bigcup_{j=1}^{q} G_{i_j,j} &= \bigcup_{j=1}^{q} \{j\} \cup \bigcup_{j=1}^{q} E_{i_j} \cup \bigcup_{j=1}^{q} F_j \cup \bigcup_{j=1}^{q} \{N - j + 1\} \\ &= [1, q] \quad \cup \quad V \quad \cup \quad F \quad \cup [N - q + 1, N] \;= [1, N] \; .\end{aligned}$$

It follows that (G, N, q) is a positive instance of SSC. □

It remains to show that whenever (G, N, q) is a positive instance of SSC, $(V, \mathcal{E})$ is a positive instance of X3C. For this, we need the following lemma.

Lemma 12.

$$\forall t \in \mathbb{Z} \quad \exists (i, j) \in [1, m] \times [1, q] \qquad (G + t) \cap [1, N] \subseteq G_{i,j} + \tau_{i,j} + t \; .$$

Proof. Let $t \in \mathbb{Z}$. $G + t$ can be written as the union of the sets $G_{i,j} + \tau_{i,j} + t$ with $(i, j) \in [1, m] \times [1, q]$. Now by construction, the $G_{i,j} + \tau_{i,j}$'s, and thus, the $G_{i,j} + \tau_{i,j} + t$'s, are distant from each other of at least N positions. It follows that the intersection of $G + t$ with $[1, N]$ cannot contain some elements of two distinct $G_{i,j} + \tau_{i,j} + t$'s. □

Now assume that (G, N, q) is a positive instance of SSC. There is $T \subseteq \mathbb{Z}$ satisfying $\#T = q$ and $[1, N] \subseteq G + T$.

Lemma 13. *There are* (i_1, j_1, u_1), (i_2, j_2, u_2), …, $(i_q, j_q, u_q) \in [1, m] \times [1, q] \times \mathbb{Z}$ *such that the sets* $G_{i_1,j_1} + u_1$, $G_{i_2,j_2} + u_2$, …, $G_{i_q,j_q} + u_q$ *are pairwise distinct and form a partition of* $[1, N]$.

Proof. Let us number arbitrarily the elements of T: $T := \{t_1, t_2, \ldots, t_q\}$. Lemma 12 guarantees that, for each $k \in [1, q]$, there are $i_k \in [1, m]$ and $j_k \in [1, q]$ satisfying $(G + t_k) \cap [1, N] \subseteq G_{i_k,j_k} + \tau_{i_k,j_k} + t_k$.

Let $u_k := \tau_{i_k,j_k} + t_k$. Since $[1, N] \subseteq G + T = \bigcup_{k=1}^{q}(G + t_k)$, it follows that $[1, N] \subseteq \bigcup_{k=1}^{q}(G + t_k) \cap [1, N] \subseteq \bigcup_{k=1}^{q}(G_{i_k,j_k} + u_k)$. So, $[1, N]$, whose cardinality is $N = q \times (2q + 4)$, is covered by the $G_{i_k,j_k} + u_k$'s (for $k \in [1, q]$), which are at most q and have each cardinality $2q + 4$. This requires that the $G_{i_k,j_k} + u_k$'s are pairwise distinct and partition $[1, N]$. □

Proving that $u_1 = u_2 = \cdots = u_q = 0$ will enable us to deduce from Lemma 13 that the G_{i_k,j_k}'s (for $k \in [1, q]$) are pairwise disjoint, and so will the q hyperedges E_{i_1}, E_{i_2}, …, E_{i_q} (be pairwise disjoint). This will mean that $\left\{E_{i_1}, E_{i_2}, \ldots, E_{i_q}\right\}$ is a partition of V, and $(V, \mathcal{E})$ a positive instance of X3C.

Let us first show that

$$\forall k \in [1, q] \qquad -q < u_k < q \,. \tag{1}$$

The integers j and $N - j + 1$ are respectively the smallest and largest elements of $G_{i,j}$. Then for any $k \in [1, q]$, one has:

$$\min(G_{i_k,j_k} + u_k) = j_k + u_k \quad \text{and} \quad \max(G_{i_k,j_k} + u_k) = N - j_k + 1 + u_k \,.$$

As $G_{i_k,j_k} + u_k$ is included in $[1, N]$ (Lemma 13), it yields $1 \le j_k + u_k$ and $N - j_k + 1 + u_k \le N$, which implies $1 - j_k \le u_k \le j_k - 1$. As j_k is at most q, one gets $1 - q \le u_k \le q - 1$, what we wanted.

Second, let us prove

$$\{j_1, j_2, \ldots, j_q\} = [1, q] \,. \tag{2}$$

By definition of j_k (Lemma 13), one has $\{j_1, j_2, \ldots, j_q\} \subseteq [1, q]$. Thus, it suffices to show that the j_k's ($k \in [1, q]$) are pairwise distinct. The proof relies on the following claim:

Claim 11. *If S is a segment of length $2q - 1$ and if u is an integer satisfying $-q < u < q$ then the center of S (*i.e.*, $(\max S + \min S)/2$) belongs to $S + u$.*

Assume there are k, $l \in [1, q]$ satisfying $k \neq l$ and $j_k = j_l$. By (1), one has $-q < u_k, u_l < q$, and so, by Claim 11, both $F_{j_k} + u_k$ and $F_{j_l} + u_l$ contain the center of $F_{j_k} = F_{j_l}$. This contradicts the fact that $G_{i_k,j_k} + u_k$ and $G_{i_l,j_l} + u_l$ are disjoint (by Lemma 13) and thus, we have shown (2).

Equation (2) allows to renumber the triples (i_k, j_k, u_k) (for $k \in [1, q]$) in such a way that $j_k = k$ for all $k \in [1, q]$.

Now, assume the set $K := \{k \in [1, q] : u_k \neq 0\}$ is non-empty, and set $\kappa := \min K$. The following claim will lead to a contradiction.

Claim 12. *Let $S \subseteq \mathbb{Z}$ and $X \subseteq S$ such that* $\min X = \min S$ *and* $\max X = \max S$. *Then X is the unique translate of X included in S (*i.e.*, for any $u \in \mathbb{Z}$, $X + u \subseteq S$ implies $u = 0$).*

For any $j \in [1, \kappa - 1]$, the set $G_{i_j, j} + u_j$

- contains j and $N - j + 1$ (since $j \notin K$ requires $u_j = 0$ and $G_{i_j,j} = G_{i_j,j} + u_j$)
- and, has an empty intersection with $G_{i_\kappa,\kappa} + u_\kappa$ (by Lemma 13).

Thus, none of j and $N - j + 1$ belongs to $G_{i_\kappa,\kappa} + u_\kappa$. As by Lemma 13, $G_{i_\kappa,\kappa} + u_\kappa$ is a subset of $[1, N]$, one gets $G_{i_\kappa,\kappa} + u_\kappa \subseteq [\kappa, N - \kappa + 1]$. Applying Claim 12 with $X := G_{i_\kappa,\kappa}$ and $S := [\kappa, N - \kappa + 1]$ yields $u_\kappa = 0$, which contradicts $\kappa \in K$.

We have then demonstrated that $K = \emptyset$, *i.e.*, that $u_1 = u_2 = \cdots = u_k = 0$. This concludes the proof of Theorem 1. □

Theorem 2. Non Detection *is* NP*-complete.*

Proof. Non Detection is in NP, since for any positive instance (g, m, k) of Non Detection, an (m, k)-similarity not detected by g is a polynomial certificate for Non Detection on (g, m, k). Hence, to obtain the NP-completeness of Non Detection, it suffices to reduce SSC to Non Detection (Theorem 1).

Let (G, N, q) be an instance of SSC. If needed, we may translate G such that $\min G = 0$; from now on we make this assumption. Thus, we have $G \subseteq [0, \max G]$. Let g be the word over $\{\texttt{\#}, \texttt{-}\}$ of length $\max G + 1$ defined by: for all j in $[1, |g|]$, $g[j] = \texttt{\#}$ iff $|g| - j \in G$.

One has $|g| - 1 = \max G \in G$ and $|g| - |g| = 0 \in G$; thus, $g[1] = g[|g|] = \texttt{\#}$, *i.e.*, the first and last letters of g are #. Let $m := N - 1 + |g|$ and $k := \min\{m, q\}$. We obtain an instance (g, m, k) of Non Detection in a time polynomial in function of (G, N, q).

Additionally, one has $N - 1 = m - |g|$ and thus,

$$[0, N - 1] = [0, m - |g|] \ . \tag{3}$$

• Assume (g, m, k) is a positive instance of Non Detection.

There is an (m, k)-similarity s that is not detected by g. Let us set $T := \{j \in [1, m] : s[j] = \texttt{0}\} - |g|$. On one hand, T is a translate of a set of cardinality k (by Definition 2) and has itself cardinality $k \leq q$. Let $i \in [0, N - 1]$. On the other hand by Equation (3), one has $i \in [0, m - |g|]$ and then, by hypothesis, g does not detect s at position i. Therefore, there exists $j \in [1, |g|]$ satisfying $g[j] = \texttt{\#}$ and $s[i + j] = \texttt{0}$. So, one gets $|g| - j \in G$ and $i + j - |g| \in T$, and this yields $i = (|g| - j) + (i + j - |g|) \in G + T$. We have thus shown that $G + T$ includes $[0, N - 1]$, from which we deduce that (G, N, q) is a positive instance of SSC.

• Conversely, let (G, N, q) be a positive instance of SSC.

Then, there exists $T \subseteq \mathbb{Z}$ having cardinality q and such that $[0, N - 1] \subseteq G + T$. Let $s \in \{\texttt{0}, \texttt{1}\}^m$ be defined by: for all $i \in [1, m]$, $s[i] = \texttt{0}$ iff $i \in T + |g|$.

Let $i \in [0, m - |g|]$. By Equation (3), one has $i \in [0, N - 1]$ and then, by hypothesis, there are $\gamma \in G$ and $t \in T$ such that $i = \gamma + t$. Setting $j := |g| - \gamma$, one gets $g[j] = \texttt{\#}$ since $|g| - j = \gamma \in G$. It follows that

$$i + j = (\gamma + t) + (|g| - \gamma) = t + |g| \in T + |g|$$

and thus, that $s[i + j] = \mathtt{0}$. It implies that g cannot detect s at position i. As i can be chosen arbitrarily in $[0, m - |g|]$, g does not detect s.

Now, it is true that $|s|_{\mathtt{0}} \leq |s| = m$ and $|s|_{\mathtt{0}} \leq \#(T + |g|) = \#T = q$, and so $|s|_{\mathtt{0}} \leq \min\{m, q\} = k$. By replacing enough 1 in s by 0's, one obtains an (m, k)-similarity that is undetected by g. It follows that (g, m, k) is a positive instance of Non Detection and this concludes the proof of Theorem 2. □

3 Hardness and Inapproximability of MWLS

In order to demonstrate the inapproximability of MWLS, we reduce Maximum Independent Set (MIS) to it. In MIS, given a graph $G = (V, \mathcal{E})$, one searches for the largest independent set I of G. It is known [10] that MIS cannot be approximated within bound $(\#V)^{0.5-\epsilon}$ unless P = NP.

Let $n \geq 1$ be an integer. Let us set $\delta_i^n := (i - 1)n^2 + i^2$ for any $i \in [1, n]$. Recall that a Golomb ruler is a set of integers such that the difference between any two distinct points in this set characterizes these two points [2].

Lemma 1. *The set $\{\delta_1^n, \delta_2^n, \ldots, \delta_n^n\}$ is a Golomb ruler with n marks computable in polynomial time in n.*

Proof. It is clear that the set $\{\delta_1^n, \delta_2^n, \ldots, \delta_n^n\}$ is computable in polynomial time in n. Let i_1, j_1, i_2, $j_2 \in [1, n]$ satisfying $i_1 < j_1$ and $i_2 < j_2$. It remains to show that our set is a Golomb ruler, *i.e.*, that $\delta_{j_1}^n - \delta_{i_1}^n = \delta_{j_2}^n - \delta_{i_2}^n$ implies $i_1 = i_2$ and $j_1 = j_2$.

For any $\alpha \in \{1, 2\}$, set $N_\alpha := \delta_{j_\alpha}^n - \delta_{i_\alpha}^n$, $q_\alpha := j_\alpha - i_\alpha$, and $r_\alpha := j_\alpha^2 - i_\alpha^2$. One has $N_\alpha = q_\alpha n^2 + r_\alpha$ and $0 \leq r_\alpha < n^2$, and so q_α and r_α are respectively the quotient and the remainder of the Euclidean division of N_α by n^2. Moreover, i_α and j_α can be written in function of q_α and r_α:

$$i_\alpha = \left(r_\alpha q_\alpha^{-1} - q_\alpha\right)/2 \qquad \text{and} \qquad j_\alpha = \left(r_\alpha q_\alpha^{-1} + q_\alpha\right)/2 \,. \tag{4}$$

Assume $\delta_{i_1}^n - \delta_{j_1}^n = \delta_{i_2}^n - \delta_{j_2}^n$. One gets $N_1 = N_2$, and by the uniqueness of the quotient and remainder of a division, one obtains $q_1 = q_2$ and $r_1 = r_2$. So, one deduces from (4) that $i_1 = i_2$ and $j_1 = j_2$. □

Definition 4 (Gadgets). *Let $X \subseteq [1, n]$. Let w_X^n denote the word over $\{\mathtt{\#}, \mathtt{-}\}$ satisfying: $|\mathrm{w}_X^n| = n^3 + n^2$, $\|\mathrm{w}_X^n\| = \#X$, and $\mathrm{w}_X^n[\delta_x^n] = \mathtt{\#}$ for any $x \in X$. Let g_X^n denote the seed obtained from w_X^n by deleting the leading and trailing - symbols.*

Next Lemma means that the g_X^n (for $X \subseteq [1, n]$) are in one-to-one correspondence with the subsets of $[1, n]$. It builds on Lemma 1.

Lemma 2. *Let X_1 and X_2 be two subsets of $[1, n]$ having cardinality at least 2. Then, $\mathrm{g}_{X_1}^n = \mathrm{g}_{X_2}^n$ if and only if $X_1 = X_2$.*

Proof. For any $\alpha \in \{1,2\}$, let us set $g_\alpha := \mathrm{g}^n_{X_\alpha}$ and $w_\alpha := \mathrm{w}^n_{X_\alpha}$. There exists $p_\alpha \in \left[0, n^3+n^2-|g_\alpha|\right]$ such that $w_\alpha = (\texttt{-})^{p_\alpha} g_\alpha (\texttt{-})^{n^3+n^2-|g_\alpha|-p_\alpha}$.

Assume $g_1 = g_2$ and let us show that $X_1 = X_2$. Notice that $X_\alpha = \{x \in [1,n] : w_\alpha[\delta^n_x] = \texttt{\#}\}$, so X_α is completely determined by w_α; therefore, it suffices to show that $w_1 = w_2$ or equivalently that $p_1 = p_2$.

One has $|g_\alpha| \geq \|g_\alpha\| = \#X_\alpha \geq 2$, and for any $i \in [1, |g_\alpha|]$, $w_\alpha[p_\alpha + i] = g_\alpha[i]$. Especially, if $i = 1$, one gets $w_\alpha[p_\alpha + 1] = g_\alpha[1] = \texttt{\#}$; so, there exists $i_\alpha \in X_\alpha$ such that $p_\alpha + 1 = \delta^n_{i_\alpha}$. Also, if $i = |g_\alpha|$, one obtains $w_\alpha[p_\alpha + |g_\alpha|] = g_\alpha[|g_\alpha|] = \texttt{\#}$ and thus, there is $j_\alpha \in X_\alpha$ satisfying $p_\alpha + |g_\alpha| = \delta^n_{j_\alpha}$.

On one hand, one has $\delta^n_{j_\alpha} = p_\alpha + |g_\alpha| \geq p_\alpha + 2 > p_\alpha + 1 = \delta^n_{i_\alpha}$. On the other hand, one also has $\delta^n_{j_1} - \delta^n_{i_1} = |g_1| - 1 = |g_2| - 1 = \delta^n_{j_2} - \delta^n_{i_2}$. Then Lemma 1 ensures that $\delta^n_{i_1} = \delta^n_{i_2}$, from which we deduce $p_1 = \delta^n_{i_1} - 1 = \delta^n_{i_2} - 1 = p_2$. □

Definition 5 (Some more gadgets). *Let $v \in [1,n]$. Let σ^n_v denote the similarity satisfying: $|\sigma^n_v| = n^3 + n^2$, $|\sigma^n_v|_{\texttt{1}} = n - 1$, and $\sigma^n_v[\delta^n_x] = \texttt{1}$ for any $x \in [1,n]$ such that $x \neq v$.*

Next Lemma explains the role of the σ^n_v's (for $v \in [1,n]$). Combined with the preceding lemma, it implies that the seeds detecting σ^n_v are in one-to-one correspondence with the g^n_X's (for $X \subseteq [1,n]$, $v \notin X$), as well as with the subsets of $[1,n]$ that do not contain v.

Lemma 3. *Let $v \in [1,n]$ and let g be a seed. Then, g detects σ^n_v if and only if there exists $X \subseteq [1,n]$ such that $v \notin X$ and $g = \mathrm{g}^n_X$.*

Proof. • Assume there is $X \subseteq [1,n]$ such that $v \notin X$ and $g = \mathrm{g}^n_X$. As g^n_X is a substring of w^n_X, it is enough to show that w^n_X detects σ^n_v at position 0. Let $i \in \left[1, n^3 + n^2\right]$ such that $\mathrm{w}^n_X[i] = \texttt{\#}$. There exists $x \in X$ such that $i = \delta^n_x$. Since $v \notin X$, one has $x \neq v$ so, $\sigma^n_v[i] = \sigma^n_v[\delta^n_x] = \texttt{1}$, what we wanted.

• Conversely, suppose g detects σ^n_v. Let $p \in [0, |\sigma^n_v| - |g|]$ such that g detects σ^n_v at position p. Then, $w := (\texttt{-})^p g (\texttt{-})^{n^3+n^2-|g|-p}$ detects σ^n_v at position 0. Let us set $X := \{x \in [1,n] : w[\delta^n_x] = \texttt{\#}\}$. First, note that $\sigma^n_v[\delta^n_v] = \texttt{0}$; consequently, $w[\delta^n_v] = \texttt{-}$ and thus, $v \notin X$. Moreover, since w detects σ^n_v and has the same length as σ^n_v, it is easy to see that $w = \mathrm{w}^n_X$ and thus, $g = \mathrm{g}^n_X$. □

Theorem 3. MWLS *is* NP*-hard. Moreover, if* MWLS *admits a polynomial time approximation algorithm with bound $(\#S)^{0.25-\epsilon}$ then* P = NP.

Proof. We reduce MIS to MWLS in such a way that it preserves the approximation properties. Let $G = (V, \mathcal{E})$ be a graph; G is an instance of MIS. Let $n := \#V$. After numbering the vertices of G, we can assume $V = [1,n]$ and thus, for any edge $E \in \mathcal{E}$, we have $E = \{\min E, \max E\}$. We build the set of similarities $\mathrm{S}_G := \{\texttt{1}^{n^3+n^2}\} \cup \{\mathrm{s}^n_E : E \in \mathcal{E}\}$ where $\mathrm{s}^n_E := \sigma^n_{\min E} \texttt{0}^{n^3+n^2} \sigma^n_{\max E}$ for any egde $E \in \mathcal{E}$. S_G is an instance of MWLS that can be constructed from G in polynomial time. Next two Lemmas guarantee that our reduction preserves the approximation.

Lemma 31. *For any independent set I of G, there is a seed of weight $\#I$ that detects all similarties in* S_G.

Proof. Let I be an independent set of G. Clearly, g_I^n is a seed of weight $\#I$ detecting $\mathtt{1}^{n^3+n^2}$. Moreover, any edge $E \in \mathcal{E}$ admits an extremity v such that $v \notin I$. Hence, by Lemma 3, g_I^n detects σ_v^n and, all the more reason for g_I^n to detect its superstring s_E^n. □

Lemma 32. *For any seed g of weight at least 2 detecting all similarities in* S_G, *there is an independent set I of G whose cardinality equals $\|g\|$. Moreover, I is computable in polynomial time in function of g.*

Proof. Let $E \in \mathcal{E}$. Let f_E be a substring of s_E^n detected by g with the same length as g. Since g starts and ends by a `#`, f_E starts and ends by a `1`. Moreover, the presence of $\mathtt{1}^{n^3+n^2}$ in S_G implies $|f_E| = |g| \leq n^3 + n^2$. Hence, the block $\mathtt{0}^{n^3+n^2}$ that lies between $\sigma_{\min E}^n$ and $\sigma_{\max E}^n$ in s_E^n is longer than f_E. This requires f_E to be fully included in $\sigma_{\min E}^n$ or $\sigma_{\max E}^n$. Thus, there exists $v_E \in \{\min E, \max E\}$ such that g detects $\sigma_{v_E}^n$ and, by Lemma 3, this garantees the existence of $X_E \subseteq [1, n]$ such that $g = \mathrm{g}_{X_E}^n$ and $v_E \notin X_E$.

Since $\#X_E = \|g\| \geq 2$, Lemma 2 implies that the X_E's (with $E \in \mathcal{E}$) are all equal to each other, and thus, their common value, denoted I, is an independent set of G of cardinality $\|g\|$. Besides, it is easy to see that $I = \{x \in [1, n] : g[\delta_x^n] = \mathtt{\#}\}$ can be computed in polynomial time from g. □

One has $\#\mathrm{S}_G = \#\mathcal{E} + 1 \leq (\#V)^2$; so, if there exists an approximation algorithm for MWLS with bound $(\#S)^{0.25-\epsilon}$, Lemmas 31 and 32 would allow to design an approximation algorithm for MIS whose bound is $(\#\mathrm{S}_G)^{0.25-\epsilon} \leq \left((\#V)^2\right)^{0.25-\epsilon} = (\#V)^{0.5-2\epsilon}$. But, this is possible only if P = NP [10]. This concludes the proof of Theorem 3. □

4 Hardness and Inapproximability of RSOS

We obtain the result on the hardness to approximate RSOS by reducing Maximum Coverage (MC) to RSOS. Our reduction is different than the one in [12] since it works even for a single seed. We use an alternative formulation of MC: given a hypergraph $(V, \mathcal{E})$ and an integer $k \geq 0$, search for a subset $C \subseteq V$ of cardinality k that maximizes the number of hyperedges $E \in \mathcal{E}$ satisfying $C \cap E \neq \emptyset$. This problem is not approximable within $\frac{e}{e-1} - \epsilon$ unless P = NP [8]. Actually, we obtain a stronger result that is the pendant to the one of Feige [8] for MC: unless P = NP, it does not exist a polynomial algorithm that, for any instance of RSOS, returns not a solution, but only an approximate value of the optimal solution within bound $\frac{e}{e-1} - \epsilon$ of the optimal.

Theorem 4. *Even if restricted to instances (d, p, S) such that $d = 1$,* RSOS *does not admit a polynomial time approximation algorithm with bound $\frac{e}{e-1} - \epsilon$ unless* P = NP.

Due to lack of space, the proof of Theorem 4 is not included in this extended abstract.

Acknowledgments

This work is supported by the CNRS STIC Specific Action #185 and the ACI IMPBio "REPEVOL". We thank G. Kucherov and L. Noé for introducing us to the problem Non Detection.

References

1. S. F. Altschul, W. Gish, W. Miller, E. W. Meyers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
2. W. C. Babcock. Intermodulation interference in radio systems. *Bell System Technical Journal*, 32(1):63–73, 1953.
3. S. Burkhardt, A. Crauser, P. Ferragina, H.-P. Lenhof, E. Rivals, and M. Vingron. q-gram Based Database Searching Using a Suffix Array (QUASAR). In *Third Annual International Conference on Computational Molecular Biology*, pages 77–83, Lyon, France, 11–14 April 1999. ACM Press.
4. S. Burkhardt and J. Kärkkäinen. Better filtering with gapped q-grams. *Fundamenta Informaticae*, 56(1–2):51–70, 2003.
5. A. Califano and I. Rigoutsos. FLASH: A fast look-up algorithm for string homology. In L. Hunter, D. Searls, and J. Shavlik, editors, *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology*, pages 56–64, Menlo Park, CA, USA, July 1993. AAAI Press.
6. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Monographs in Computer Science. Springer, 1999.
7. M. Farach-Colton, G. M. Landau, S. Cenk Sahinalp, and D. Tsur. Optimal spaced seeds that avoid false negatives. url: http://cs.haifa.ac.il/~landau/gadi/seeds.ps.
8. U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the Association for Computing Machinery*, 45(4):634–652, 1998.
9. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of* NP*-Completeness.* W. H. Freeman and Co., 1979.
10. J. Håstad. Clique is hard to approximate within $n^{1-\epsilon}$. *Acta Mathematica*, 182:105–142, 1999.
11. G. Kucherov, L. Noé, and M. Roytberg. Multi-seed lossless filtration. In *Proceedings of the 15h Annual Symposium on Combinatorial Pattern Matching (CPM'04)*, volume 3109, pages 297–310. Lecture Notes in Computer Science, 2004.
12. M. Li, B. Ma, D. Kisman, and J. Tromp. PatternHunter II: Highly sensitive and fast homology search. *Journal of Bioinformatics and Computational Biology*, 2(3):417–439, 2004.
13. B. Ma, J. Tromp, and M. Li. Patternhunter: faster and more sensitive homology search. *Bioinformatics*, 18(3):440–445, 2002.
14. L. Noé and G. Kucherov. Improved hit criteria for DNA local alignment. *BMC Bioinformatics*, 5(149), 2004. doi:10.1186/1471-2105-5-149.