Seed Optimization Is No Easier than Optimal Golomb Ruler Design

Bin Ma

Department of Computer Science
University of Western Ontario
London, ON, N6A5B7, Canada
E-mail: bma@csd.uwo.ca

Hongyi Yao

Institute for Theoretical Computer Science
Tsinghua University
Beijing, 100084, China
E-mail: thy03@mails.tsinghua.edu.cn

Spaced seed is a filter method invented to efficiently identify the regions of interest in similarity searches. It is now well known that certain spaced seeds hit (detect) a randomly sampled similarity region with higher probabilities than the others. Assume each position of the similarity region is identity with probability p independently. The seed optimization problem seeks for the optimal seed achieving the highest hit probability with given length and weight. Despite that the problem was previously shown not to be NP-hard, in practice it seems difficult to solve. The only algorithm known to compute the optimal seed is still exhaustive search in exponential time. In this article we put some insight into the hardness of the seed design problem by demonstrating the relation between the seed optimization problem and the optimal Golomb ruler design problem, which is a well known difficult problem in combinatorial design.

Keywords: spaced seeds; Golomb ruler; reduction

1. Introduction and Notations

1.1. *Seed optimization*

Similarity searches often utilize some types of filtrations to efficiently identify the similarity candidates for further examination. Normally filtration provides a tradeoff between searching sensitivity and searching speed. In DNA similarity searches, spaced seed was invented to achieve a better tradeoff.[1]

A spaced seed x is represented by a binary string such as 111*1**1*1*1**11*111. The positions with letter 1 are required matches, and the positions with letter * are "don't cares". The length of the string is called the *length* of the seed, denoted by $l(x)$. The number of required matches is called the *weight* of the seed, denoted by $w(x)$. A similarity is *hit* by a seed x if there is a length-$l(x)$ segment of the similarity such that all the required matches specified by x are satisfied by the

segment. Figure 1 shows an example.

```
GAGTACTCAACACCAACATTAGTGGGCAATGGAAAAT
|| ||||||||| ||||| || ||||||   ||||||
GAATACTCAACAGCAACACTAATGGGCAGCAGAAAAT
       111*1**1*1**11*111
```

Fig. 1. The seed 111*1**1*1**11*111 hits the similarity region.

A spaced seed x can also be specified by the set of positions of the required matches. For example, the seed $x = 111*1**1*1**11*111$ can be denoted by its *set representation* $S(x) = \{0, 1, 2, 4, 7, 9, 12, 13, 15, 16, 17\}$. For a given set S and an integer, we define $S + i = \{x + i \mid x \in S\}$.

It is easy to see that spaced seeds with the same weight provide the same efficiency in filtering out random matches. However, it was observed, though not thoroughly studied, that some spaced seeds provide better filtration than the others.[2,3] Ma et al.[1] first studied the optimization of the spaced seed in their PatternHunter paper, and demonstrated that the optimized spaced seed could improve the sensitivity (hit probability) significantly over the consecutive seed (with no * in the seed) of the same weight. The term, spaced seed, was also coined in the paper.[1]

In the PatternHunter paper,[1] a length-L similarity region is modeled as a 0-1 string, where 0 means mismatch and 1 means match. Each position of the region is independently 1 with probability p. In this paper we call these regions the i.i.d. regions [*] and p be the *similarity level* of the region. Then[1] enumerated all the possible seeds with given weight and length, calculated their hit probabilities under certain L and p, and selected the optimal seed with the best hit probability. This is apparently an exponential time algorithm. After several years of extensive research in the seed optimization problem, many heuristic algorithms were developed to calculate the optimal seed.[5,7–12] However, the exponential-time, brute-force algorithm is still the only known algorithm that guarantees the finding of the optimal seed. We formalize the seed optimization problem under i.i.d. regions as follows:

I.I.D. Seed Optimization An instance of i.i.d. seed optimization is given by a four-tuple $\langle l, w, L, p\rangle$. The objective is to find the seed with length l and weight w that achieves the maximum hit probability in i.i.d. regions with length L and similarity level p.

Clearly, the similarity regions have another simple probabilistic model, where a length-L similarity region is uniformly drawn from all length-L 0-1 strings with exactly k matches (letter 1). In this paper we call these regions the uniform regions. Analogously, we define the following:

Uniform Seed Optimization An instance of uniform seed optimization is

[*] Different from this paper,[4,5] used the term "uniform" for these regions. The use of the terms i.i.d. and uniform in this paper follow.[6]

given by a four-tuple $\langle l, w, L, k\rangle$. The objective is to find the seed with length l and weight w that achieves the maximum hit probability in uniform regions with length L and exactly k matches.

Independently to the work of PatternHunter,[1] Burkhardt and Kärkkäínen[13] studied a slightly different seed optimization under uniform regions. They tried to find a seed with the maximum weight to hit *all* the uniform regions. Apparently, this problem can be reduced to the uniform seed optimization problem by trying different values of l and w.

Despite the hardness of seed optimization in practice, Li et al.[4,14] made an interesting observation that if the input parameters are given in unary forms, then the seed optimization problem can not be NP-hard. This observation is based on the theorem that a sparse language (the number of instances is bounded by a polynomial of the input size) cannot be NP-hard unless P=NP.[15] Thus, the research in seed optimization is in an awkward situation: no efficient algorithm has been designed; yet NP-hardness, the common strategy to prove the complexities of a problem, does not work here.

Much related to the seed optimization problems, researchers have studied the algorithms to calculate the sensitivity of *a given spaced seed*, under both the i.i.d. and uniform models. Under the i.i.d. model, Ma et al.[1] proposed the first exponential time algorithm and other papers[16,17] proposed algorithms with improved time complexity. Under the uniform model, Buhler et al.[12] proposed exponential time algorithm. The sensitivity calculation algorithms have been used in the brute-forth seed optimization as subroutines. Hence, sensitivity calculation appeared to be an easier problem than seed optimization. Ironically, the accurate sensitivity calculation was proved to be NP-hard under both the i.i.d. model[4,14] and the uniform model.[18]

However, the proofs of hardness of sensitivity calculation do not imply the hardness of the seed optimization. This is because the proofs required specially designed spaced seeds, which may not be the optimal seeds. In this paper we aim to provide some insight into the complexity of the seed optimization problem.

1.2. *Golomb ruler*

A w-mark Golomb ruler is a set of distinct nonnegative integers $0 = a_1, a_2, \ldots, a_w$, called "marks", such that $|a_i - a_j| \neq |a_k - a_l|$ for $\{i, j\} \neq \{k, l\}$ and $i \neq j$. The optimal Golomb ruler design problem seeks for a w-mark ruler with the least a_w.[19]

It is relatively easy to construct a w-mark Golomb ruler with polynomial a_w. In fact, because of the easy construction, Golomb ruler has been used in the reduction to prove the NP-hardness of calculating the sensitivity of a given spaced seed.[4,18]

However, the finding of the optimal Golomb ruler is much harder. Although there is no mathematical proof about the computational complexity of optimal Golomb ruler design, it is well known in combinatorial design that optimal Golomb ruler design is a very difficult problem. The largest known optimal Golomb ruler to date

has $w = 24$, which was found by J. P. Robinson and A. J. Bernstein[20] in 1967 and verified to be optimal with four years of distributed computation at distributed.net (http://www.distributed.net) in 2004. Currently the finding (verifying) of the 25-mark optimal Golomb ruler is underway at distributed.net.

The optimal Golomb ruler design problem and our seed optimization problem are analogous in the simplicity of the definitions and the complexity of the algorithms in use. Indeed, in this paper we reduce the optimal Golomb ruler design problem to seed optimization, and consequently prove that seed optimization is at least as hard as optimal Golomb ruler design. Our results, together with the tremendous efforts that mathematicians have spent on optimal Golomb ruler design, justify the exponential time algorithms and heuristic algorithms for seed optimization, and suggest that the future research in this problem should still focus on these two types of algorithms.

The rest of the paper is organized as follows: Section 2 proves that in the i.i.d. regions with certain conditions, optimal seeds are Golomb rulers. A closed-form sufficient condition is given. This reduces the optimal Golomb ruler design problem to the i.i.d. seed optimization problem. Section 2 further provides a counterexample to show that without the conditions, the optimal Golomb ruler may not be the optimal seed. Section 3 studies the uniform seed optimization. Results in uniform regions are very similar to the i.i.d. regions. Section 4 discusses the results and proposes open problems.

2. I.I.D. Seed Optimization

2.1. *Reduction from optimal Golomb ruler design to i.i.d. seed optimization*

In this section we provide a polynomial time reduction from the optimal Golomb ruler design problem to the seed optimization problem. It has been believed that the sensitivity increase of the spaced seed comes from the irregularities in the positions of the letters 1 in the seed. With the irregularity, when a spaced seed hits a similarity region, an extra hit right after the first hit requires many more positions of the similarity region to be matches, as illustrated in Figure 2. This makes the concurrent

```
111*1**1*1**11*111        111*1**1*1**11*111         ...
 111*1**1*1**11*111         111*1**1*1**11*111       ...
```

Fig. 2. No matter how the seed is "slided", two overlapping 111*1**1*1**11*111 always give six or more extra required matches than one seed.

existence of more than one hits in the same similarity region a rare event; whereas for a consecutive seed, the second hit is relatively easy – only one additional required match is needed. As a result, while the total number of hits are similar, spaced seeds hit more similarity regions than a consecutive seed.

Noticing that if seed x is such that its set representation, $S(x)$ is a Golomb ruler, then $S(x) \cap (S(x) + i)$ has at most one element for any integer i. This provides the minimum level of overlap between a seed and its sliding. For the above mentioned relation between sensitivity and irregularity, a Golomb ruler is likely to the optimal spaced seed. This is not necessarily true for all conditions (Section 2.2). However, in what follows we prove that this is true under certain conditions.

We first give a very stringent condition in Theorem 2.1. Later on this condition will be relaxed in Theorem 2.2.

Theorem 2.1. *Consider the i.i.d. seed optimization problem $\langle l, w, L, p\rangle$. Let $n = L - l + 1$ be the number of positions the seed can hit the region. Suppose $p \leq \frac{1}{n^3}$ and $n \geq 2l$. Then there is a w-mark Golomb ruler with $a_w = l - 1$ if and only if the optimal spaced seed is a Golomb ruler.*

Proof. Suppose a length-l and weight-w seed is given by its set representation. When the context is clear, we also use S to refer to the seed. Define $\phi(i) = |S \cap (S + i)|$. Define $\phi = \max_i \phi(i)$.

Denote by $h(i_1, \ldots, i_k)$ the probability of that the seed hits at every position of i_1, i_2, …, i_k. This event is equivalent to that all the positions in

$$\bigcup_{j=1}^{k} (S + i_k)$$

are matches. Therefore, it is easy to verify that Equations (1), (2) and (3) are true.

For any $0 \leq i < n$,

$$h(i) = p^w. \tag{1}$$

For any $0 \leq i < j < n$,

$$h(i, j) = p^{2w-\phi(j-i)} \leq p^{2w-\phi}. \tag{2}$$

For any $0 \leq i < j < k < n$,

$$h(i, j, k) \leq p^{2w-\phi+1}. \tag{3}$$

We claim that

$$Pr(S \text{ } hits) \geq \sum_{i=0}^{n-1} h(i) - \sum_{0 \leq i < j < n} h(i, j) \tag{4}$$

and

$$Pr(S \text{ } hits) \leq \sum_{i=0}^{n-1} h(i) - \sum_{0 \leq i < j < n} h(i, j) + \sum_{0 \leq i < j < k < n} h(i, j, k) \tag{5}$$

This is because of the following two facts: (1) for any similarity region that contains $r \leq 2$ hits, the probability of the region is counted precisely once in both Eq.(4) and Eq.(5); (2) for any similarity region that contains $r > 2$ hits, the probability of

the region is counted $\binom{r}{1} - \binom{r}{2} \leq 1$ time in Eq.(4) and $\binom{r}{1} - \binom{r}{2} + \binom{r}{3} \geq 1$ times in Eq.(5).

Because of Eq.(3), when $p \leq \frac{1}{n^3}$,

$$\sum_{0 \leq i < j < k < n} h(i,j,k) \leq \binom{n}{3} p^{2w-\phi+1} < p^{2w-\phi} \times \frac{1}{2} \tag{6}$$

If $\phi = 1$, Eq.(4) becomes

$$Pr(Shits) \geq np^w - p^{2w-1} \times \frac{n^2}{2} \tag{7}$$

If $\phi \geq 2$, because there is at least one pair of i and j such that $\phi(j-i) = \phi \geq 2$, as well as Eq.(6), Eq.(5) becomes

$$Pr(S\ hits) \leq np^w - p^{2w-\phi} + p^{2w-\phi} \times \frac{1}{2} = np^w - \frac{1}{2} \times p^{2w-\phi} < np^w - p^{2w-1} \times \frac{n^2}{2} \tag{8}$$

When there is a Golomb ruler of length l with w markers, the seed defined by the ruler has $\phi = 1$ and the hit probability is lower bounded by (7). Because $\phi \geq 2$ implies (8), the optimal seed must be such that $\phi = 1$. It is easy to verify that when $n \geq 2l$, $\phi = 1$ implies that the seed is a Golomb ruler. □

Corollary 2.1. *The i.i.d. seed optimization problem is at least as hard as optimal Golomb ruler design.*

Proof. Theorem 2.1 says that the finding of w-mark Golomb ruler with length l can be reduced to the seed optimization problem. Then the optimal Golomb ruler problem for a given weight w can be solved by trying different length l in polynomial steps. □

One problem of Theorem 2.1 is that the upper bound of p is $O(n^{-3})$, which is very small and not practical. We relax this upper bound in Theorem 2.2.

Theorem 2.2. *Consider the i.i.d. seed optimization problem $\langle l, w, L, p \rangle$. Let $n = L - l + 1$. Suppose $p \leq \frac{1}{4l} \cdot \frac{n-l}{n}$ and $2l \leq n \leq (2\sqrt{l})^{w-1}$. Then there is a w-mark Golomb ruler with $a_w = l - 1$ if and only if each optimal spaced seed is a Golomb ruler.*

Proof. If there is no optimal w-mark Golomb ruler with $a_w = l - 1$, then clearly the optimal spaced seed cannot be a Golomb ruler. Next we prove the "only if". Suppose there is a w-mark Golomb ruler, denoted as S^*. Denote the optimal spaced seed as S. We prove by contradiction that S is also a Golomb ruler.

Define $\phi^*(i) = |S^* \cap (S^* + i)|$. Because S^* is a Golomb ruler, $\phi^*(i) \leq 1$. Define $\phi(i) = |S \cap (S + i)|$ and $\phi = \max_i \phi(i)$. If S is not a Golomb ruler, then $\phi > 1$. Let $h^*(i_1, \ldots, i_k) = Pr(S^*\ hits\ at\ i_1, \ldots, i_k)$ and $h(i_1, \ldots, i_k) = Pr(S\ hits\ at\ i_1, \ldots, i_k)$.

Because both S and S^* have weight w, $h(i) = h^*(i) = p^w$. In addition, if $j - i \geq l$, then $h(i,j) = h^*(i,j) = p^{2w}$. Thus, replacing S by S^* and h by h^* in Eq.(4), then subtracting Eq.(5) from Eq.(4), we get the following:

$$\begin{aligned}
& Pr(S^* \ hits) - Pr(S \ hits) \\
\geq & \sum_{0 \leq i < j < n} h(i,j) - \sum_{0 \leq i < j < n} h^*(i,j) - \sum_{0 \leq i < j < k < n} h(i,j,k) \\
= & \sum_{0 \leq i < j < \min(i+l,n)} h(i,j) - \sum_{0 \leq i < j < \min(i+l,n)} h^*(i,j) - \sum_{0 \leq i < j < k < n} h(i,j,k) \\
\geq & (n-l)p^{2w-\phi} - nlp^{2w-1} - \sum_{0 \leq i < j < k < n} h(i,j,k). \qquad (9)
\end{aligned}$$

Here the last inequality is because the following two facts: (1) There is at least one d such that $\phi(d) = \phi$. Therefore $h(i, i+d) = p^{2w-\phi}$ for at least $n - l$ different i. (2) $h^*(i,j) \leq p^{2w-1}$.

To prove the theorem, it suffices to show that when $\phi \geq 2$, Eq. (9) is greater than zero, which is a contradiction to the optimality of S. Clearly, when p is small, the absolute value of the second negative term in (9) can be bounded by a fraction of the first term in (9). We need to examine the third term $\sum_{0 \leq i < j < k < n} h(i,j,k)$ more carefully.

The set of indexes $I = \{(i,j,k) | 0 \leq i < j < k < n\}$ can be divided into two sets $I_1 = \{(i,j,k) \in I | j < i + l \text{ and } k < j + l\}$ and $I_2 = I \setminus I_1$. Clearly I_1 corresponds to the situations where the seed at j overlaps both the seed at i and the seed at k; and I_2 corresponds to the situations where at least one hit does not overlap the other two seeds. For any $(i,j,k) \in I_2$, $|(S+i) \cup (S+j) \cup (S+k)| \geq 3w - \phi$. Hence

$$\sum_{(i,j,k) \in I_2} h(i,j,k) \leq \frac{n^3}{6} \times p^{3w-\phi}. \qquad (10)$$

When p is small, this can also be bounded by a fraction of the first term of (9).

Again, I_1 can be divided into two sets

$$J_1 = \{(i,j,k) \in I_1 \ : \ |(S+k) \setminus ((S+i) \cup (S+j))| = 1\}$$

and $J_2 = I_1 \setminus J_1$. That is, providing that there are hits at i and j, J_1 contains the indexes where the the seed at k requires only one additional match in the similarity region, and J_2 contains the indexes where the seed at k requires at least two additional matches. Therefore, for any $(i,j,k) \in J_2$, $|(S+i) \cup (S+j) \cup (S+k)| \geq 2w - \phi + 2$. Hence

$$\sum_{(i,j,k) \in J_2} h(i,j,k) \leq nl^2 \times p^{2w-\phi+2}. \qquad (11)$$

When p is small, this can be bounded by a fraction of the first term of (9) again. The rest of the proof is to bound

$$\sum_{(i,j,k) \in J_1} h(i,j,k) \leq p^{2w-\phi+1} \times |J_1|.$$

For $(i, j, k) \in J_1$, we consider the possibilities of k for fixed i and j. As shown in Figure 3, because the last letter 1 in the seed at k already contributed an additional match, the second last letter 1 in the seed at k must coincide with the last letter 1 of either the seed at i or the seed at j. Otherwise it contributes another additional match (keep in mind that the seeds at i, j and k are the same seed), contradicting the definition of J_1.

```
i:    ...10001                    ...10001
j:      ...10001                    ...10001
k:          ...10001                    ...10001

          (A)                         (B)
```

Fig. 3. The two possible choices of k for fixed i and j in J_1.

Thus, once i is fixed, there are l choices of j, and then there are at most two choices of k. As a result, $|J_1| \leq 2nl$, and

$$\sum_{(i,j,k)\in J_1} h(i,j,k) \leq p^{2w-\phi+1} \times 2nl \tag{12}$$

Combining Equations (9), (10), (11) and (12),

$$\begin{aligned}
&Pr(S^* \ hits) - Pr(S \ hits) \\
&> (n-l)p^{2w-\phi} - nlp^{2w-1} - \sum_{(i,j,k)\in I_2\cup J_2\cup J_1} h(i,j,k) \\
&\geq (n-l)p^{2w-\phi} - nlp^{2w-1} - \frac{n^3}{6}\cdot p^{3w-\phi} - nl^2\cdot p^{2w-\phi+2} - 2nl\cdot p^{2w-\phi+1} \\
&= p^{2w-\phi+1}\left((n-l)\cdot p^{-1} - nl\cdot p^{\phi-2} - \frac{n^3}{6}\cdot p^{w-1} - nl^2\cdot p - 2nl\right)
\end{aligned}$$

It is easy to verify that when $\phi \geq 2$, $p \leq \frac{1}{4l}\cdot\frac{n-l}{n}$ and $n \leq (2\sqrt{l})^{w-1}$, the above equation has value greater than 0. Hence the theorem is proved. □

Remark: Obviously, the main factor for the upper bound of p in Theorem 2.2 is $O(\frac{1}{l})$. With a more sophisticated analysis, it is also possible to bound p by $O(\frac{1}{w})$. The analysis is omitted here.

2.2. *Counterexample*

One natural question is to ask whether the upper bounds on p in Theorem 2.1 and Theorem 2.2 can be removed. The answer is no. With much computation, we found the following counter example. For $w = 5$, $n = 150$, $p = 0.999$, the optimal 5-mark Golomb ruler is $\{0, 2, 7, 10, 11\}$.[21] The corresponding spaced seed has sensitivity

$1 - 4.3376 \times 10^{-116}$. Whereas the spaced seed $\{0, 3, 4, 6, 11\}$, which is not a Golomb ruler, has a better sensitivity $1 - 3.3674 \times 10^{-117}$.

One may argue that this may be due to that the region is not sufficiently long, and the boundary effects cause this to happen. Figure 4 excludes this possibility. When $p = 0.999$, the curves in Figure 4 plot the trend of $\log(Pr(There\ is\ no\ hit))$ for the above two seeds, with respect to the length of the region. Clearly we can see that the no-hit probability of the Golomb ruler seed $\{0, 2, 7, 10, 11\}$ goes down slower than the non-Golomb ruler seed $\{0, 3, 4, 6, 11\}$. This clearly demonstrates that the non-Golomb ruler seed $\{0, 3, 4, 6, 11\}$ is asymptotically better than the Golomb ruler seed.

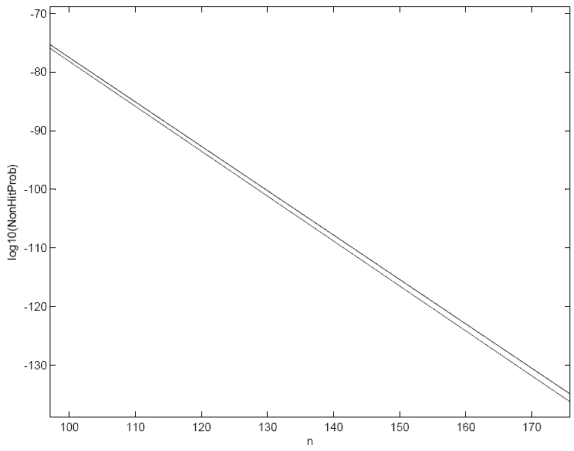

Fig. 4. The curves of $\log(Pr(There\ is\ no\ hit))$ with respect to the region length. The upper curve is for the Golomb ruler seed $\{0, 2, 7, 10, 11\}$, and the lower curve is for the seed $\{0, 3, 4, 6, 11\}$.

3. Uniform Seed Optimization

For uniform regions, we can also reduce the optimal Golomb ruler design problem to seed optimization. The proof is in fact much simpler.

Theorem 3.1. *Optimal Golomb ruler design can be reduced to uniform seed optimization problem in linear time.*

Proof. Suppose the uniform regions are given with length L and k matches. Let $k = 2w - 2$ and $L \geq 3l$.

For any seed S, let $\phi(i)$ as defined in the proof of Theorem 2.1. Clearly,

$$Pr(S\ hits) \leq \sum_{i=0}^{L-l} Pr(S\ hits\ at\ i).$$

Furthermore, the equality holds if and only if $Pr(S \textit{ hits more than once}) = 0$. With $k = 2w - 2$, this happens if and only if $\phi(i) \leq 1$ for any i, i.e., S is a Golomb ruler seed. □

Similar to the i.i.d. regions, the conditions on the similarity level k cannot be removed. The counterexample for i.i.d. region still works here. When $L = 200$, $k = 140$, $l = 12$, and $w = 5$, the non-Golomb ruler seed $0, 3, 4, 6, 11$ has better sensitivity $1 - 1.34 \times 10^{-36}$ than the Golomb ruler seed $0, 2, 7, 10, 11$, of which the sensitivity is $1 - 9.3 \times 10^{-35}$.

4. Discussion and Open Problems

Although seed optimization was proved not to be NP-hard in the literature, in this paper we put some insight into its computational complexity by a polynomial time reduction from another well-known difficult problem, the optimal Golomb ruler design. In fact, we show that *under certain conditions*, the following statement holds:

Statement: If a Golomb ruler exists, the optimal seed is a Golomb ruler.

However, without those conditions, we also give a counterexample to show that the statement is not always true. In fact, our counterexample shows that a non-Golomb ruler seed can be asymptotically better than the optimal Golomb ruler seed. This is different from a common belief in seed design that the irregularities in the seeds increase the seed sensitivity. Our example shows that the factors that determine the seed sensitivity is more involved than just the irregularity.

For i.i.d. regions, the conditions for the statement to be true are mainly on the similarity level, p, of the similarity region. The best upper bound we give in the paper is $p = O(\frac{1}{l})$, where l is the length of the desired seed. For uniform regions, the condition is much more stringent: k must be equal to $2w - 2$. We leave it as an open question whether a significantly more relaxed condition on p or k exist for the statement to hold for i.i.d. or uniform regions.

Our seed optimization problem is given in the form of $\langle l, w, L, p\rangle$ for i.i.d. regions and $\langle l, w, L, k\rangle$ for uniform regions. In practice often the length of the seed, l, is not fixed for optimizing a seed. When l is not fixed, our reduction does not straightforwardly imply the complexity of seed optimization. This is because the optimal seed can possibly have a shorter length than the w-mark optimal Golomb ruler, as a consequence of the simple fact that shorter seeds have more positions to hit a length-L region. However, if the i.i.d. seed optimization problem is defined to maximize the hit probability at the first n positions of a region, then the length of a seed is not important anymore and our results still hold. We also point out that all our reductions work for circular regions, with or without the parameter l.

Although our results indicate that the seed optimization problem is very hard (at least as hard as optimal Golomb ruler design), whether a polynomial time algorithm exists for seed optimization is still an open problem.

5. Acknowledgment

The work was supported in part by China NSF 60553001, National Basic Research Program of China 2007CB807900,2007CB807901, NSERC and Canada Research Chair. BM's work was done when he visited Prof. Andrew Yao at ITCS at Tsinghua University. BM thanks Dr. Ming Li and Dr. John Tromp for commenting on an earlier version of this manuscript. The authors thank the useful discussion with the attendees of a seminar course at ITCS at Tsinghua University. In particular, Yifei Zhang pointed out the Golomb ruler design problem during the course.

References

1. B. Ma, J. Tromp and M. Li, *Bioinformatics* **18**, 440.
2. P. Pevzner and M. Waterman, *Algorithmica* **13**, 135 (1995).
3. O. Lehtinen, E. Sutinen and J. Tarhio, Experiments on block indexing, in *Proc. of the 3rd South American Workshop on String Processing*, 1996.
4. M. Li, B. Ma and L. Zhang, Superiority and complexity of the spaced seeds, in *Proc. of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.
5. M. Li, B. Ma, D. Kisman and J. Tromp, *J. Bioinf. and Comp. Biol.* **2(3)**, 417 (2004).
6. M. Csürös and B. Ma, *Algorithmica* **48**, 187 (2007).
7. J. Xu, D. Brown, M. Li and B. Ma, Optimizing multiple spaced seeds for homology search, in *Proc. of the 15th Symposium on Combinatorial Pattern Matching (CPM), LNCS 3109*, 2004.
8. L. Ilie and S. Ilie, Fast computation of good multiple spaced seeds, in *Proc. of 7th Workshop on Algorithms in Bioinformatics*, 2007.
9. K. P. Choi, F. Zeng and L. Zhang, *Bioinformatics* **20**, 1053 (2004).
10. F. Preparata, L. Zhang and K. Choi, *J. Comput. Biol.* **12**, 137 (2005).
11. I.H.Yang, S.H.Wang, H. Chen, P. Huang and K.M.Chao, Efficient methods for generating optimal single and multiple spaced seeds, in *Proc. of IEEE 4th Symp. on Bioinformatics and Bioengineering*, 2004.
12. J. Buhler, U. Keich and Y. Sun, Designing seeds for similarity search in genomic DNA, in *Proc. of the 7th International Conference on Computational Biology (RECOMB)*, 2003.
13. S. Burkhardt and J. Kärkkäínen, *Fundamenta Informaticae* **23**, 1001 (2003).
14. B. Ma and M. Li, *Journal of Computer Science and System Sciences* (2007).
15. S. Mahaney, *Journal of Computer and System Sciences* **25**, 130 (1982).
16. U. Keich, M. Li, B. Ma and J. Tromp, *Discrete Appl. Math* **3**, 253 (2004).
17. K. Choi and L. Zhang, *J. Comp and Sys. Sci.* **68** (2004), 22–40.
18. F. Nicolas and E. Rivals, Hardness of optimal spaced seed design, in *Proc. 16th Annual Symposium Combinatorial Pattern Matching (CPM'05)*, 2005.
19. C. J. Colbourn and J. H. Dinitz (eds.), *CRC Handbook of Combinatorial Designs* (Boca Raton, FL: CRC Press, 1996).
20. J. Robinson and A. Bernstein, *IEEE Trans. Inform. Th.* **13**, 106 (1967).
21. A. Dollas, W. Rankin and D. McCracken, *IEEE Transaction on Information Theory* **44**, 379 (1998).