

AN ASYMPTOTIC LOWER BOUND FOR THE MAXIMAL NUMBER OF RUNS IN A STRING

FRANTISEK FRANEK

*Department of Computing and Software, McMaster University
Hamilton, Ontario L8S 4K1, Canada*

and

QIAN YANG

*Department of Computing and Software, McMaster University
Hamilton, Ontario L8S 4K1, Canada*

Received (received date)

Revised (revised date)

Communicated by Editor's name

ABSTRACT

An asymptotic lower bound for the maxrun function $\rho(n) = \max \{ \text{number of runs in string } \mathbf{x} \mid \text{all strings } \mathbf{x} \text{ of length } n \}$ is presented. More precisely, it is shown that for any $\varepsilon > 0$, $(\alpha - \varepsilon)n$ is an asymptotic lower bound, where $\alpha = \frac{3}{1+\sqrt{5}} \approx 0.927$. A recent construction of an increasing sequence of binary strings “rich in runs” is modified and extended to prove the result.

Keywords: run, lower bound, asymptotic lower run, maximum number of runs

1. Introduction

An important structural characteristic of a string over an alphabet is its periodicity. Repetitions (tandem repeats) have always been in the focus of the research into periodicities. The notion of runs captures maximal repetitions which themselves are not repetitions and allows for a succinct notation ([1]). Even though it had been known that there could be $O(n \log n)$ of repetitions in a string of length n ([2]), it was shown in 2000 by Kolpakov and Kucherov that number of runs was linear in the length of the input string ([3]). Their proof did not allow to specify the constants of linearity. The behaviour of the **maxrun function** $\rho(n) = \max\{\text{number of runs in string } \mathbf{x} \mid \text{all strings } \mathbf{x} \text{ of length } n\}$ became an interest of study to many. Smyth et al. (e.g. [4], [5], [6]) presented a set of conjectures about $\rho(n)$:

1. $\rho(n) < n$,
2. or in an asymptotically stronger version: $\lim_{n \rightarrow \infty} \frac{\rho(n)}{n}$ exists and equals $\frac{3}{1+\sqrt{5}}$,

3. $\rho(n+1) \leq \rho(n)+2$,
4. $\rho(n)$ is attained on a binary cube-free string of length n .

Recently, there has been a flurry of results concerning the upper bound of $\rho(n)$: first Rytter set the upper bound of $\rho(n)$ to $5n$ (see [7]), which was subsequently improved by Puglisi, Simpson, and Smyth to $3.48n$ ([8]) and also by Rytter himself to $3.44n$. Recently, Crochemore and Ilie ([9]) pushed the upper bound down to $1.6n$, indicating that a further computer analysis can obtain an upper bound as low as $1.18n$.

Franek, Simpson, and Smyth in [4] introduced a recursive construction of a sequence $\{\mathbf{x}_n : n < \infty\}$ of binary strings increasing in length and “rich in runs” so that $\lim_{n \rightarrow \infty} \frac{r(\mathbf{x}_n)}{|\mathbf{x}_n|} = \alpha$, where $\alpha = \frac{3}{1+\sqrt{5}} \approx 0.927$ and $r(\mathbf{x}) = \text{number of runs in } \mathbf{x}$. Although any such sequence does not establish a lower bound (not even an asymptotic one), it has been “viewed” as such. The assumption underneath that view is that $\rho(n)$ behaves “reasonably”, i.e. that $\rho(n)$ does not exhibit wild jumps up, or equivalently, that $\frac{\rho(n)}{n}$ does not exhibit wild oscillations, which is generally expected to be the case (cf. the second conjecture). However, since the “reasonable behaviour” of $\rho(n)$ is yet to be established, we modify and extend the method from [4] to provide formally a family of true asymptotic lower bounds arbitrarily close to αn by proving

Theorem: *For any $\varepsilon > 0$ there is a positive integer N so that for any $n \geq N$, $\rho(n) \geq (\alpha - \varepsilon)n$.*

2. Basic notation, facts, and methods

A **run** \mathcal{R} in a string \mathbf{x} is a four-tuple of positive integers (s, p, e, t) , where

1. s is the **starting position** of \mathcal{R} .
2. p is the **size of its period**.
3. $e \geq 2$ is its **exponent**, i.e. the maximal value e so that $\mathbf{x}[s..s+p-1] = \mathbf{x}[s+p..s+2p-1] = \dots = \mathbf{x}[s+(e-1)p..s+ep-1]$.
4. The **period** of \mathcal{R} , $\mathbf{x}[s..s+p-1]$ itself is not a repetition.
5. The **square part** of the run \mathcal{R} , $\mathbf{x}[s..s+p-1] = \mathbf{x}[s+p..s+2p-1]$ is **left-maximal**, i.e. $\mathbf{x}[s-1..s+p-2] \neq \mathbf{x}[s+p-1..s+2p-2]$.
6. t is the **tail** of \mathcal{R} and indicates how far to the right the run can be extended, i.e. t is a maximal number so that for any $0 < t' \leq t$, $\mathbf{x}[s+t'..s+t'+p-1] = \mathbf{x}[s+t'+p..s+t'+2p-1] = \dots = \mathbf{x}[s+t'+(e-1)p..s+t'+ep-1]$.

Not too much is known about the behaviour of the maxrun function:

- For any n , $\rho(n+2) \geq \rho(n)+1$.

Take a string \mathbf{x} of length n with $r(\mathbf{x}) = \rho(n)$. Take a letter c that does not occur in \mathbf{x} . Then $\mathbf{x}cc$ is a string of length $n+2$ and $\rho(n+2) \geq r(\mathbf{x}cc) = r(\mathbf{x})+1 = \rho(n)+1$.

- For any n , $\rho(n+1) \leq \rho(n) + \lfloor \frac{n}{2} \rfloor$.

Take a string \mathbf{x} of length $n+1$ with $r(\mathbf{x}) = \rho(n+1)$. There can be at most $\lfloor \frac{n}{2} \rfloor$ squares starting at position 1. Then $\rho(n) \geq r(\mathbf{x}[2..n+1]) \geq r(\mathbf{x}) - \lfloor \frac{n}{2} \rfloor \geq \rho(n+1) - \lfloor \frac{n}{2} \rfloor$.

- For some n , $\rho(n+1) = \rho(n)$.

Established by computations, it is not clear if this as an asymptotic property (for instance, $\rho(33) = 27$ while $\rho(34) = 27$).

- For some n , $\rho(n+1) = \rho(n)+2$.

Established by computations, it is not clear if this as an asymptotic property (for instance, $\rho(13) = 8$ while $\rho(14) = 10$).

Note that the function $\frac{\rho(n)}{n}$ may not be monotonic. It is not even clear whether $\lim_{n \rightarrow \infty} \frac{\rho(n)}{n}$ exists, as $\frac{\rho(n)}{n}$ may be oscillating with a non-decreasing magnitude.

In [4] a special concatenation operator \circ for binary strings was introduced:

$$\mathbf{x}[1..n] \circ \mathbf{y}[1..m] = \begin{cases} \mathbf{x}[1..n]\mathbf{y}[2..m] = \mathbf{x}[1..n-1]\mathbf{y}[1..m] & \text{if } \mathbf{x}[n] = \mathbf{y}[1], \\ \mathbf{x}[1..n-1]\mathbf{y}[2..m] & \text{if } \mathbf{x}[n] \neq \mathbf{y}[1]. \end{cases}$$

Morphism g was defined by

$$g(\mathbf{x}) = \begin{cases} 010010 & \text{if } \mathbf{x} = 0, \\ 101101 & \text{if } \mathbf{x} = 1, \\ g(\mathbf{x}[1..n]) = g(\mathbf{x}[1]) \circ g(\mathbf{x}[2]) \circ \dots \circ g(\mathbf{x}[n]) & \text{if } |\mathbf{x}| > 1 \end{cases} \quad (1)$$

The strings 010010 and 101101 were selected as they provide in the concatenation one extra run:

$r(g(0) \circ g(0)) = 6 = 2r(g(0))+2$, the same for $g(1) \circ g(1)$, $r(g(0) \circ g(1)) = 5 = r(g(0))+r(g(1))+1$, the same for $r(g(1) \circ g(0))$. Let us remark that a computer search carried to the length of 20 did not discover any better pair of strings with such properties.

An important aspect of the morphism is that it “preserves” the runs in \mathbf{x} : it is a bit tedious to prove and thus not included in the paper, but any left-maximal square in \mathbf{x} induces a square in $g(\mathbf{x})$. It follows that every run in \mathbf{x} induces a run in $g(\mathbf{x})$. It is also important to show that two distinct runs in \mathbf{x} do not get “glued” together by g .

Let us fix a string \mathbf{x} . Let $\lambda(\mathbf{x})$ denote the number of pairs 00 or 11 in \mathbf{x} . We can calculate the length of $g(\mathbf{x})$:

$$|g(\mathbf{x})| = 6|\mathbf{x}| - \lambda(\mathbf{x}) - 2(|\mathbf{x}| - \lambda(\mathbf{x}) - 1) = 4|\mathbf{x}| + \lambda(\mathbf{x}) + 2 \quad (2)$$

the number of pairs 00 or 11 in $g(\mathbf{x})$:

$$\lambda(g(\mathbf{x})) = |\mathbf{x}| \quad (3)$$

the number of runs in $g(\mathbf{x})$:

$$r(g(\mathbf{x})) = r(\mathbf{x}) + 2|\mathbf{x}| + (|\mathbf{x}| - 1) = r(\mathbf{x}) + 3|\mathbf{x}| - 1 \quad (4)$$

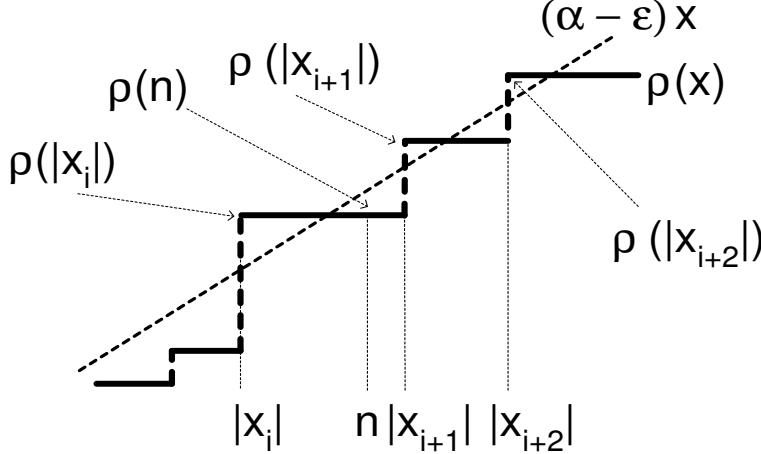


Fig. 1. $\rho(n)$ function between $|\mathbf{x}_i|$ and $n|\mathbf{x}_{i+1}|$

In [4] a sequence of strings was generated inductively from a starting string, for instance: $\mathbf{x}_0 = 0$, $\mathbf{x}_1 = g(0) = 010010$, and $\mathbf{x}_{i+1} = g(\mathbf{x}_i)$ for $i \geq 1$. Then $|\mathbf{x}_{i+1}| = 4|\mathbf{x}_i| + |\mathbf{x}_{i-1}| + 2$ according (2) and $r(\mathbf{x}_{i+1}) = r(\mathbf{x}_i) + 3|\mathbf{x}_i| - 1$ according to (4). It is not hard to show that the limit $\lim_{i \rightarrow \infty} \frac{|\mathbf{x}_i|}{|\mathbf{x}_{i+1}|}$ exists and $\beta = \lim_{i \rightarrow \infty} \frac{|\mathbf{x}_i|}{|\mathbf{x}_{i+1}|} = -2 + \sqrt{5}$. The limit $\lim_{i \rightarrow \infty} \frac{r(\mathbf{x}_i)}{|\mathbf{x}_i|}$ also exists and $\alpha = \lim_{i \rightarrow \infty} \frac{r(\mathbf{x}_i)}{|\mathbf{x}_i|} = \beta(\alpha + 3)$ giving $\alpha = \frac{3}{1 + \sqrt{5}}$.

The sequence $\{|\mathbf{x}_i| : i < \infty\}$ is only ‘‘probing’’ the domain of the function $\rho(n)$ and $r(\mathbf{x}_i)$ is ‘‘pushing’’ the value of $\rho(n)$ above αn in these ‘‘probing’’ points (see Figure 2). Since the size of \mathbf{x}_{i+1} is more than 4 times the size of \mathbf{x}_i , the gaps between $|\mathbf{x}_i|$ and $|\mathbf{x}_{i+1}|$ are getting bigger and bigger.

The basic idea of establishing an asymptotic lower bound for $\rho(n)$ (see Figure 2) is to start similar sequences from various ‘‘starting’’ strings to cover the domain of $\rho(n)$ densely enough with the ‘‘probing’’ points to get any n close to some ‘‘probing’’ point and hence the value of $\rho(n)$ close to αn . To be able to do so, we must change a bit the way the sequences are generated. The details of this are in the next section. The nature of the next section is by necessity very technical – it only deals with the mathematical machinery of constructing an asymptotic lower bound from several sequences.

3. The proof of the theorem

Let $\varepsilon > 0$ be given. We have to find N so that for any $n \geq N$, $\rho(n) \geq (\alpha - \varepsilon)n$.

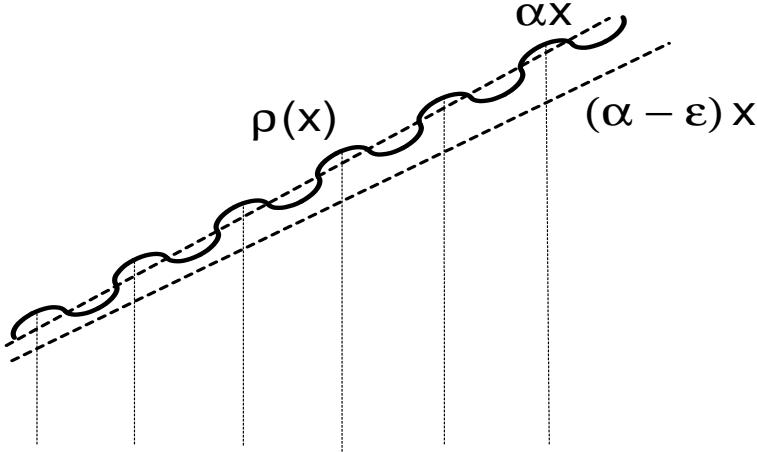


Fig. 2. The idea of construction of a lower bound

The proof is conducted in several stages.

Stage 1: selection of parameters

First we choose and fix three parameters k , δ , and R that will be used throughout the whole proof. These parameters depend on the given ε and are selected so that at the end $\rho(n) \leq (\alpha - \varepsilon)n$.

We choose and fix a positive integer k so that $\frac{\alpha}{k+1} < \varepsilon$; then we choose and fix a positive real δ so that $\delta \leq \frac{k+1}{k}(\varepsilon - \frac{\alpha}{k+1})$. It follows that $\frac{k}{k+1}(\alpha - \delta) \geq \alpha - \varepsilon$. Then for R we choose the smallest integer so that $(\frac{k+1}{k})^R \geq 5$.

Stage 2: introduction of special $\mathcal{S}_{a,b}$ sequences.

Consider an increasing sequence $\mathcal{S}_{a,b}$ of positive integers with two integer parameters a and b defined by

$$n_0(a, b) = a,$$

$$n_1(a, b) = 4a + b, \text{ and}$$

$$n_{i+2}(a, b) = 4n_{i+1}(a, b) + n_i(a, b) \text{ for } i \geq 0.$$

It is not hard to show that $\lim_{i \rightarrow \infty} \frac{n_i(a, b)}{n_{i+1}(a, b)}$ exists and that

$$\lim_{i \rightarrow \infty} \frac{n_i(a, b)}{n_{i+1}(a, b)} = -2 + \sqrt{5}.$$

Importantly, ranges of such sequences are “tied” together based on the parameters, i.e. for any integer $t \geq 1$ and any i

$$n_i(ta, tb) = tn_i(a, b). \quad (5)$$

Stage 3: defining parameters $a(j)$ and $b(j)$

For $0 \leq j < R$, set

$$a(j) = 3(k+1)^j k^{(R-j)} \text{ and } b(j) = \frac{a(j)}{3} = (k+1)^j k^{(R-j)}. \quad (6)$$

It follows that $\frac{k+1}{k}a(j) = a(j+1)$, $\frac{k+1}{k}b(j) = b(j+1)$, and $b(j) \geq 3$.

Stage 4: defining new morphism $h(\mathbf{x})$ for binary strings

Based on the morphism $g(\mathbf{x})$ (see (1)) we define a new morphism $h(\mathbf{x})$ by removing the last 2 letters from $g(\mathbf{x})$:

$$\text{if } g(\mathbf{x}) = \mathbf{y}[1..n], \text{ then } h(\mathbf{x}) = \mathbf{y}[1..n-2] \quad (7)$$

It follows that

$$|h(\mathbf{x})| = 4|\mathbf{x}| + \lambda(\mathbf{x}) \quad (8)$$

while

$$\lambda(g(\mathbf{x})) = \lambda(h(\mathbf{x})) = |\mathbf{x}| \quad (9)$$

remains unchanged.

We need to estimate the number of runs in $h(\mathbf{x})$. In general, it is not clear how many runs in $g(\mathbf{x})$ can be destroyed by removing the last two letters of $h(\mathbf{x})$. But, since we will be only dealing with a special kind of strings (we will call them *good*), we can make an estimate of how many runs we might be losing.

We use the term *string s ends with a square* to indicate that s has a left-maximal square as its suffix. We call a string *good* if it has length ≥ 4 , has 011 as its suffix, and ends with at most one square that is different from the suffix 11.

- Claim:* (a) if \mathbf{x} is good, then $h(\mathbf{x})$ is good
(b) if \mathbf{x} is good, then $r(g(\mathbf{x})) \geq r(h(\mathbf{x})) \geq r(g(\mathbf{x})) - 2$.

To simplify this argument, the claim will be proven after completing this whole proof.

Stage 5: defining the “probing” sequences of strings

For any $0 \leq j < R$ we define a sequence of binary strings $\{\mathbf{x}_i(j) : i < \infty\}$ by:

$$\mathbf{x}_0(j) = (011)^{b(j)}$$

and for any $i \geq 0$,

$$\mathbf{x}_{i+1}(j) = h(\mathbf{x}_i(j))$$

where $b(j)$ is defined in (6). From (2) and (4) it follows that for any $i \geq 0$,

$$|\mathbf{x}_0(j)| = 3b(j) = a(j),$$

$$|\mathbf{x}_1(j)| = 4a(j) + b(j), \text{ and}$$

$$|\mathbf{x}_{i+2}(j)| = 4|\mathbf{x}_{i+1}(j)| + |\mathbf{x}_i(j)|.$$

Thus, the sequence $\{|\mathbf{x}_i(j)| : i < \infty\}$ is an $S_{a(j), b(j)}$ sequence and therefore $\lim_{i \rightarrow \infty} \frac{|\mathbf{x}_i(j)|}{|\mathbf{x}_{i+1}(j)|} = -2 + \sqrt{5}$.

Since our starting string $\mathbf{x}_0(j)$ is good as it equals $(011)^{b(j)}$ and $b(j) \geq 3$, according to the **Claim**, every $\mathbf{x}_i(j)$ is good, and

$$\mathbf{r}(g(\mathbf{x}_i(j))) \geq \mathbf{r}(\mathbf{x}_{i+1}(j)) \geq \mathbf{r}(g(\mathbf{x}_i(j))) - 2$$

and so

$$\lim_{i \rightarrow \infty} \frac{\mathbf{r}(\mathbf{x}_i(j))}{|\mathbf{x}_i(j)|} = \alpha.$$

Therefore, for any $0 \leq j < R$ there is a positive integer I_j so that for any $i \geq I_j$,

$$\frac{\rho(|\mathbf{x}_i(j)|)}{|\mathbf{x}_i(j)|} \geq \frac{\mathbf{r}(\mathbf{x}_i(j))}{|\mathbf{x}_i(j)|} \geq \alpha - \delta.$$

Let $I = \max\{I_j : 0 \leq j < R\}$. Then for any $i \geq I$ and any $0 \leq j < R$,

$$\frac{\rho(|\mathbf{x}_i(j)|)}{|\mathbf{x}_i(j)|} \geq \frac{\mathbf{r}(\mathbf{x}_i(j))}{|\mathbf{x}_i(j)|} \geq \alpha - \delta. \quad (10)$$

From (5) and (6) it follows, that for any i and any $0 \leq j < R$,

$$n_i(a(j), b(j)) = \left(\frac{k+1}{k}\right) n_i(a(j-1), b(j-1)) = \dots = \left(\frac{k+1}{k}\right)^j n_i(a(0), b(0)).$$

Set $N = \max\{n_I(a(j), b(j)) : 0 \leq j < R\}$. This is the N we were searching for.

If $n \geq N$, then for some $i \geq I$,

$$n_i(a(0), b(0)) < n \leq n_{i+1}(a(0), b(0)).$$

Then there is $0 \leq j < R-1$ so that

$$\left(\frac{k+1}{k}\right)^j n_i(a(0), b(0)) < n \leq \left(\frac{k+1}{k}\right)^{j+1} n_i(a(0), b(0))$$

[since $\left(\frac{k+1}{k}\right)^R \geq 5$, then $\left(\frac{k+1}{k}\right)^R n_i(a(0), b(0)) \geq n_{i+1}(a(0), b(0))$].

It follows that

$$n_i(a(j), b(j)) < n \leq \frac{k+1}{k} n_i(a(j), b(j)).$$

Now we can estimate the value of $\frac{\rho(n)}{n}$ using (10):

$$\frac{\rho(n)}{n} \geq \frac{\rho(n_i(a(j), b(j)))}{n} \geq \frac{k}{k+1} \frac{\rho(n_i(a(j), b(j)))}{n_i(a(j), b(j))} \geq \frac{k}{k+1} (\alpha - \delta) \geq \alpha - \varepsilon.$$

Thus $\rho(n) \geq (\alpha - \varepsilon)n$. \square

In the above argument we skipped proving the **Claim** in order not to complicate the already complicated proof of the main theorem. We are restating the **Claim** here and providing its proof.

Claim: (a) if \mathbf{x} is good, then $h(\mathbf{x})$ is good
(b) if \mathbf{x} is good, then $r(g(\mathbf{x})) \geq r(h(\mathbf{x})) \geq r(g(\mathbf{x})) - 2$.

Proof. Let \mathbf{x} be good. Then $\mathbf{x} = \mathbf{u}011$ and so $g(\mathbf{x}) = g(\mathbf{u}) \circ 010010 \circ 101101 \circ 101101 = g(\mathbf{u}) \circ 010010110101101$. It follows that $h(\mathbf{x}) = g(\mathbf{u}) \circ 0100101101011$. Thus $h(\mathbf{x})$ ends with 011.

If \mathbf{x} ends with a square different from the suffix 11 (let us call it \mathcal{S}), then $g(\mathbf{x})$ ends with two squares: the one resulting from \mathcal{S} and 101|101 (resulting from 11). Removing the last two letters 01 of $g(\mathbf{x})$ will destroy both of them. The run 001011|01011|01 in $g(\mathbf{x})$ will not be destroyed, it will become a leftmost square in $h(\mathbf{x})$. Therefore, $h(\mathbf{x})$ will end with at most one square different from the suffix 11. It follows that $h(\mathbf{x})$ is good and (a) is proven.

Moreover, we saw that at most two squares in $g(\mathbf{x})$ were destroyed. Hence (b) is verified as well. \square

4. Conclusion and further research

We showed that the expectation of αn being a lower bound for the maxrun function $\rho(n)$ is valid by proving that there is a whole family of asymptotic lower bounds arbitrarily close to αn . Qian Yang in her master thesis ([10]) presented a different recursive construction of a sequence of binary strings of an increasing lengths so that $\lim_{n \rightarrow \infty} \frac{r(\mathbf{x}_n)}{|\mathbf{x}_n|} = \alpha$. This additional result and the fact that the lower bound for $\rho(n)$ had been pushed so low strengthen the evidence for the conjecture that $\lim_{n \rightarrow \infty} \frac{\rho(n)}{n}$ exists and equals $\frac{3}{1+\sqrt{5}}$.

The further research will include trying to push the lower bound higher up to see whether the conjecture $\rho(n) < n$ holds. This will involve finding novel ways of creating strings “rich in runs” as the approach with concatenation \circ seems to give as much as it could.

Acknowledgements

The research was supported in part by the first authors’ research grant from the Natural Sciences and Engineering Research Council of Canada.

References

1. M.G. Main, “Detecting leftmost maximal periodicities”, *Discr. Applied Maths.* 25 (1989) 145-153.
2. M. Crochemore, “An optimal algorithm for computing the repetitions in a word”, *Inform. Process. Lett.*, 12-5 (1981) 297–315
3. R. Kolpakov and G. Kucherov, “On maximal repetitions in words”, *J. Discrete Algorithms*, 1 (2000) 159-186.
4. F. Franek, J. Simpson, and W.F. Smyth, “The maximum number of runs in a string”, *Proc. AWOCA 2003, 14th Australasian Workshop on Combinatorial Algorithms*, Seoul National University, Seoul, Korea, 13 - 16 July, 2003
5. S.J. Puglisi, W.F. Smyth, and A. Turpin, “Some restrictions on Periodicity in Strings”, *Proc. of AWOCA 2005, 16th Australasian Workshop on Combinatorial*

Algorithms, University of Ballarat, Victoria, Australia, September 18-21, 2005, 263-268.

6. Kangmin Fan and W.F. Smyth, “A new periodicity lemma”, to appear in *SIAM J. of Discr. Math.*
7. W. Rytter, “The Number of Runs in a String: Improved Analysis of the Linear Upper Bound”, *Proc. of STACS 2006, 23rd Annual Symposium on Theoretical Aspects of Computer Science*, Marseille, France, February 23-25, 2006 184-195
8. S.J. Puglisi, J. Simpson, and W.F. Smyth, “How many runs can a string contain?”, submitted.
9. M. Crochemore and L. Ilie, “Maximal repetitions in strings”, submitted.
10. Q. Yang, “Lower and upper bounds for maximum number of runs” *M.Sc. thesis*, Dept. of Comp. and Software, McMaster University, January 2007