McMaster University

Advanced Optimization Laboratory

Title:

A parameterized formulation for the maximum number of runs problem

Authors:

Andrew Baker[1], Antoine Deza[1,2], and Frantisek Franek[1]

[1] Advanced Optimization Laboratory,
Department of Computing and Software,
McMaster University, Hamilton, Ontario, Canada.
[2] Equipe Combinatoire et Optimisation,
Université Pierre et Marie Curie, Paris, France
Email: bakerar2, deza, franek@mcmaster.ca

AdvOL-Report No. 2011/2

March 2011, Hamilton, Ontario, Canada

A parameterized formulation for the maximum number of runs problem *

Andrew Baker, Antoine Deza, Frantisek Franek

March 24, 2011

Abstract

A parameterized approach to the problem of the maximum number of runs in a string was introduced by Deza and Franek. In the approach referred to as the *d-step approach*, in addition to the usual parameter the length of the string, the size of the string's alphabet is considered. The behaviour of the function $\rho_d(n)$, the maximum number of runs over all strings of length n with exactly d distinct symbols, can be handily expressed in the terms of properties of a table referred to as the $(d, n-d)$ *table* in which $\rho_d(n)$ is the entry at the dth row and $(n-d)$th column. The approach leads to a conjectured upper bound $\rho_d(n) \leq n-d$ for $2 \leq d \leq n$. The parameterized formulation shows that the maximum within any column of the $(d, n-d)$ table is achieved on the main diagonal, i.e. for $n = 2d$, and motivates the investigation of the structural properties of the run-maximal strings of length n bounded by a constant times the size of the alphabet d. We show that $\rho_d(n) = \rho_{n-d}(2n-2d)$ for $2 \leq d \leq n < 2d$, $\rho_d(2d) \leq \rho_{d-1}(2d-1)+1$ for $d \geq 3$, $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2) = \rho_{d-3}(2d-3)$ for $d \geq 5$, and $\{\rho_d(n) \leq n-d$ for $2 \leq d \leq n\} \Leftrightarrow \{\rho_d(9d) \leq 8d$ for $d \geq 2\}$. The results allow for an efficient computational verification of entries in the $(d, n-d)$ table for higher values of n and point to a plausible way of either proving the maximum number of runs conjecture by showing that possible counter-examples on the main diagonal would exhibit an impossible structure, or to discover an unexpected counter-example on the main diagonal of the $(d, n-d)$ table. This approach provides a purely analytical proof of $\rho_d(2d) = d$ for $d \leq 15$ and, using the computational results of $\rho_2(d+2)$ for $d = 16, \ldots, 23$, a proof of $\rho_d(2d) = d$ for $d \leq 23$.

1 Introduction

The problem of determining the maximum number of runs in a string has a rich history and many researchers have contributed to the effort. The notion of a run is due to Main [16], the term itself was introduced in [12]. Kolpakov and Kucherov [13, 14] showed that the function $\rho(n)$, the maximum number of runs over all strings of length n, is linear. Several papers dealt with lower and upper bounds or expected values for $\rho(n)$, see [2, 3, 4, 8, 9, 10, 11, 17, 18, 19, 20, 22] and references therein.

*This work was supported by the *Natural Sciences and Engineering Research Council of Canada*, *MPRIME*, and by the *Canada Research Chair program*, and made possible by the facilities of the *Shared Hierarchical Academic Research Computing Network* (`http://www.sharcnet.ca/`).

The counting estimates leading to the best upper bounds [3, 4] rely heavily on a computational approach and seem to reach a point where it gets highly challenging, bordering intractability, to verify the results or make further progress. A few researchers tried a structural approach. Rytter's three neighbour lemma can be considered one such attempt, along with the ongoing work of W. Smyth *et al.* [6, 7, 15, 21].

A parameterized approach to the investigation of the structural aspects of run-maximal strings was introduced by Deza and Franek [5]. In addition to considering the length of the string they introduced the parameter d giving the function $\rho_d(n)$, the maximum number of runs over all strings of length n with exactly d distinct symbols. These values are presented in the so-called $(d, n-d)$ table, where the value of $\rho_d(n)$ is the entry at the row d and the column $n-d$. In Table 1, the entries for the first 10 rows and the first 10 columns are presented. Several properties of the table were presented in [5], the most important being the fact that $\rho_d(n) \leq n-d$ for $2 \leq d \leq n$ is equivalent with $\rho_d(2d) \leq d$ for $d \geq 2$. In other words, if the diagonal obeys the upper bound $n-d$, so do all the entries in the table everywhere. Though in the related literature, the *maximum number of runs conjecture* – or simply *runs conjecture* – refers to the hypothesis that $\rho(n) \leq n$, in this paper we will take it to be $\rho_d(n) \leq n-d$.

We discuss several additional properties of the $(d, n-d)$ table, the behaviour of the function $\rho_d(n)$ on or nearby the main diagonal, and discuss some structural properties of run-maximal strings on the main diagonal. The results allow for the extension of computational verification of the maximum number of runs conjecture to higher values of n and also indicate a viable approach to an analytical investigation of the conjecture by either showing a possible counter-example to the conjecture would have to exhibit an impossible structure, or exhibiting a counter-example on the main diagonal of the $(d, n-d)$ table and direct calculation of entries for smaller columns.

Let us remark, that although we believe with the majority of the researchers in the field that the conjecture is true and hence view the d-step approach as a possible tool to prove it, if a counter-example exists, one must be on the main diagonal and we believe it will easier to find there as the run-maximal strings of length being twice the size of the alphabet seem to exhibit a richer structure than general run-maximal strings. A counter-example would be in essence a quite striking result.

2 Notation and Preliminaries

Throughout this paper, we refer to k-tuples: a symbol which occurs exactly k times in the string under consideration. Specially named k-tuples are the *singleton* (1-tuple), *pair* (2-tuple), *triple* (3-tuple), *quadruple* (4-tuple), and *quintuple* (5-tuple).

Definition 2.1 *A safe position in a string $\boldsymbol{x}$ is one which, when removed from $\boldsymbol{x}$, does not result in two runs being merged into one in the resulting new string.*

A safe position does not ensure that the number of runs will not change when that position is removed, only that no runs will be lost through being merged; runs may still be destroyed by having an essential symbol removed. Safe positions are important in that they may be

Table 1: Values for $\rho_d(n)$ with $1 \leq d \leq 10$ and $1 \leq n - d \leq 10$.For more values, see [1]

		$n-d$										
		1	2	3	4	5	6	7	8	9	10	11
d	1	**1**	*1*	1	1	1	1	1	1	1	1	.
	2	1	**2**	*2*	3	4	5	5	6	7	8	.
	3	1	2	**3**	*3*	4	5	6	6	7	8	.
	4	1	2	3	**4**	*4*	5	6	7	7	8	.
	5	1	2	3	4	**5**	*5*	6	7	8	8	.
	6	1	2	3	4	5	**6**	*6*	7	8	9	.
	7	1	2	3	4	5	6	**7**	*7*	8	9	.
	8	1	2	3	4	5	6	7	**8**	*8*	9	.
	9	1	2	3	4	5	6	7	8	**9**	*9*	.
	10	1	2	3	4	5	6	7	8	9	**10**	.
	11	.	.	.	.	.	.	.	.	.	.	.

removed from a string while only affecting the runs which contain them. When the position of a symbol is unambiguous, we may thus refer to a *safe symbol* rather than to its position – for instance we can talk about a safe singleton, or about the first member of a pair being safe, etc.

At various points we will need to relabel all occurrences of a symbol in a string or substring. Let $\boldsymbol{x}_b^a$ denote the string $\boldsymbol{x}$, in which all occurrences of a are replaced by b, and vice versa. $S_d(n)$ refers to the set of strings of length n with exactly d distinct symbols. For a string $\boldsymbol{x}$, $\mathcal{A}(\boldsymbol{x})$ denotes the alphabet of $\boldsymbol{x}$, while $r(\boldsymbol{x})$ denotes the number of runs of $\boldsymbol{x}$.

Lemma 2.2 *There exists a run-maximal string in $S_d(n)$ with no unsafe singletons for $2 \leq d \leq n$.*

Proof. Let $\boldsymbol{x}$ be a run-maximal string in $S_d(n)$. We will show that one of the following conditions must hold:

- (i) $\boldsymbol{x}$ has no singletons, or
- (ii) $\boldsymbol{x}$ has exactly one singleton which is safe, or
- (iii) $\boldsymbol{x}$ has exactly one singleton which is unsafe, and there exists another run-maximal string $\boldsymbol{x'} \in S_d(n)$ where $\boldsymbol{x'}$ has no unsafe singletons, or
- (iv) $\boldsymbol{x}$ has more than one singleton, all of which are safe.

Let $\boldsymbol{x}$ have some unsafe singletons.

First, consider the case that $\boldsymbol{x}$ has exactly one singleton, C, which is unsafe: $\boldsymbol{x} = \boldsymbol{u}a\boldsymbol{v}a\boldsymbol{v}Ca\boldsymbol{v}a\boldsymbol{v}\boldsymbol{w}$, where $\boldsymbol{u}$, $\boldsymbol{v}$, and $\boldsymbol{w}$ are (possibly empty) strings, and $a \in \mathcal{A}(\boldsymbol{x}) - \{C\}$. Let $\boldsymbol{x'} = \boldsymbol{u}a\boldsymbol{v}a\boldsymbol{v}(Ca\boldsymbol{v}a\boldsymbol{v}\boldsymbol{w})_C^a =$

$\boldsymbol{uavav}(aC\boldsymbol{v}^a_C C\boldsymbol{v}^a_C\boldsymbol{w}^a_C) = \boldsymbol{uavavaC\tilde{v}C\tilde{v}\tilde{w}}$. Clearly, $\boldsymbol{x'} \in S_d(n)$,$r(\boldsymbol{x'}) \geq r(\boldsymbol{x})$, so $\boldsymbol{x'}$ is run-maximal and has no singletons.

Next, consider the case that $\boldsymbol{x}$ has at least 2 singletons C, D, of which one is unsafe, C. Without loss of generality, we can assume C occurs before D : $\boldsymbol{x} = \boldsymbol{uavavCavavwDz}$, where $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{w}$, and $\boldsymbol{z}$ are (possibly empty) strings and $a \in \mathcal{A}(\boldsymbol{x}) - \{C, D\}$. Let $\boldsymbol{x_1} = \boldsymbol{uavav}(C\boldsymbol{avavwDz})^a_C = \boldsymbol{uavavaC\tilde{v}C\tilde{v}\tilde{w}D\tilde{z}}$. Clearly, $\boldsymbol{x_1} \in S_d(n)$ and $r(\boldsymbol{x_1}) \geq r(\boldsymbol{x})$. We then modify $\boldsymbol{x_1}$ by removing the safe symbol a immediately to the left of the first occurrence of C, yielding $\boldsymbol{x_2}$. Finally, we add a second copy of D adjacent to the original D, restoring the original length: $\boldsymbol{x_3} = \boldsymbol{uavavC\tilde{v}C\tilde{v}\tilde{w}DD\tilde{z}}$. $\boldsymbol{x_3} \in S_d(n)$ and $r(\boldsymbol{x_3}) > r(\boldsymbol{x_2}) \geq r(\boldsymbol{x_1}) \geq r(\boldsymbol{x})$, which contradicts the run-maximality of $\boldsymbol{x}$. □

Lemma 2.3 is a simple observation that for a position to be unsafe, a symbol must occur twice to the left and twice to the right of that position.

Lemma 2.3 *If a string $\boldsymbol{x}$ consists only of singletons, pairs, and triples, then every position is safe.*

A corollary of Lemma 2.3 is that the maximum number of runs in a string with only singletons, pairs, and triples is limited by the number of pairs and triples. Specifically, $r(\boldsymbol{x}) = \#pairs + \lfloor \frac{3}{2} \#triples \rfloor$. This is because a pair can only be involved in a single run, and a triple can be involved in at most 2 runs. The densest structure achievable is through overlapping triples in the pattern *aababb*, which has 3 runs for every two triples. The pairs, meanwhile, are maximized through adjacent copies.

3 Run-maximal strings below the main diagonal and in the immediate neighbourhood above

We first remark that every value below the main diagonal in the $(d, n-d)$ table is equal to the value on the main diagonal directly above it. In other words, the values on and below the main diagonal in a column are constant.

Proposition 3.1 *We have $\rho_d(n) = \rho_{n-d}(2n-2d)$ for $2 \leq d \leq n < 2d$.*

Proof. Consider a run-maximal string $\boldsymbol{x} \in S_d(n)$, where $2 \leq d \leq n < 2d$. By Lemma 2.2, we can assume $\boldsymbol{x}$ has no unsafe singletons. Since $n < 2d$, $\boldsymbol{x}$ must have a singleton, and hence it must be safe. We can remove this safe singleton, yielding a new string $\boldsymbol{y} \in S_{d-1}(n-1)$ and so $\rho_d(n) = r(\boldsymbol{x}) = r(\boldsymbol{y}) \leq \rho_{d-1}(n-1)$. Recall the following inequality noted in [5]:

$$\rho_d(n) \leq \rho_{d+1}(n+1) \text{ for } 2 \leq d \leq n \tag{1}$$

Thus, $\rho_{d-1}(n-1) = \rho_d(n)$. □

Proposition 3.1 together with inequality (1) gives the following equivalency noted in [5]: $\{\rho_d(n) \leq n-d \text{ for } 2 \leq d \leq n\} \Leftrightarrow \{\rho_d(2d) \leq d \text{ for } 2 \leq d\}$.

If there is a counter-example to the conjectured upper bound, then the main diagonal must contain a counter-example. If it falls under the main diagonal, then by Proposition 3.1 there

must be a counter-example on the main diagonal – i.e. it can be *pushed up*, and if it falls above the main diagonal, by the inequality (1), there must be a counter-example on the main diagonal – i.e. the counter-example can be *pushed down*.

We extend Proposition 3.1 to bound the behaviour of the entries in the immediate neighbourhood above the main diagonal in the $(d, n-d)$ table. Proposition 3.2 establishes that the difference between the entry on the main diagonal and the entry immediately above it is at most 1. In addition, the difference is 1 if and only if every run-maximal string in $S_d(2d)$ consists entirely of pairs; otherwise, the difference is 0.

Proposition 3.2 *We have* $\rho_d(2d) \leq \rho_{d-1}(2d-1)+1$ *for* $d \geq 3$.

Proof. Let $\boldsymbol{x} \in S_d(2d)$ be a run-maximal string with no unsafe singletons (by Lemma 2.2). If $\boldsymbol{x}$ does not have a singleton, then it consists entirely of pairs. It is clear that the pairs must be adjacent and that $r(\boldsymbol{x}) = d$ and so $\boldsymbol{x} = aabbcc\ldots$. Removing the first a and renaming the second to b, $\boldsymbol{y} = bbbcc\ldots \in S_{d-1}(2d-1)$ and $\rho_{d-1}(2d-1) \geq r(\boldsymbol{y}) = r(\boldsymbol{x}) - 1 = \rho_d(2d) - 1$. If $\boldsymbol{x}$ has a singleton, since it is safe we can remove it forming a string $\boldsymbol{y} \in S_{d-1}(2d-1)$ so that $\rho_{d-1}(2d-1) \geq r(\boldsymbol{y}) = r(\boldsymbol{x}) = \rho_d(2d)$, and so $\rho_{d-1}(2d-1) = \rho_d(2d)$. □

We have seen that the gap between the first entry above the diagonal and the diagonal entry is at most 1. Proposition 3.3 establishes that the three entries just above the diagonal are identical.

Proposition 3.3 *We have* $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2) = \rho_{d-3}(2d-3)$ *for* $d \geq 5$.

Proof. Let $\boldsymbol{x}$ be a run-maximal string in $S_{d-1}(2d-1)$. By Lemma 2.2 we can assume that either it has a safe singleton or no singletons at all. In the former case, we can remove the safe singleton obtaining $\boldsymbol{y} \in S_{d-2}(2d-2)$ so that $\rho_{d-2}(2d-2) \geq r(\boldsymbol{y}) \geq r(\boldsymbol{x}) = \rho_{d-1}(2d-1)$, and so $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2)$. In the latter case, $\boldsymbol{x}$ consists of pairs and one triple, and thus, by Lemma 2.3, all positions are safe. Therefore, we can move all the pairs to the end of the string, yielding $\boldsymbol{y} = aaabbcc\ldots \in S_{d-1}(2d-1)$ and by removing the first a and renaming the remaining as to cs, $\boldsymbol{z} = ccbbcc\ldots \in S_{d-2}(2d-2)$. It follows that $\rho_{d-2}(2d-2) \geq r(\boldsymbol{z}) = r(\boldsymbol{y}) = r(\boldsymbol{x}) = \rho_{d-1}(2d-1)$, and so $\rho_{d-1}(2d-1) = \rho_{d-2}(2d-2)$.

Let $\boldsymbol{x}$ be now a run-maximal string in $S_{d-2}(2d-2)$. Again, if $\boldsymbol{x}$ has a singleton, we can assume by Lemma 2.2 it is safe and form $\boldsymbol{y}$ by removing the singleton. $\boldsymbol{y} \in S_{d-3}(2d-3)$ and $\rho_{d-3}(2d-3) \geq r(\boldsymbol{y}) \geq r(\boldsymbol{x}) = \rho_{d-2}(2d-2)$. If $\boldsymbol{x}$ does not have a singleton, then $r(\boldsymbol{x}) = d-1$. To see this, consider the two cases:

(*i*) $\boldsymbol{x}$ consists of two triples and several pairs. The most runs which may be obtained in such a string, after grouping the pairs at the end of the string, is through the arrangement $aababbccddee\ldots$. In this case, there are $d-4$ runs from the pairs, and 3 runs from the triples, giving a total of $d-1$ runs.

(*ii*) $\boldsymbol{x}$ consists of a quadruple and several pairs. The most runs which may be obtained in this case is from a string with either the structure $aabbaaccddee\ldots$, or $aabaabccddee\ldots$, where all the pairs have been grouped at the end, except for the pair of bs which is used to break up the quadruple. In both cases, there are $d-4$ runs involving characters c

onward, and three runs involving the characters a and b, again giving a total of $d-1$ runs.

Now consider a string $\boldsymbol{y} = aabbaabbcdee\ldots \in S_{d-2}(2d-2)$, which has two quadruples (of as and bs), two singletons (c and d), and several pairs ($e\ldots$). This string has $d-6$ runs from the pairs ee onward, and 5 runs from the characters a and b, giving a total of $d-1$ runs, i.e. $r(\boldsymbol{x}) = r(\boldsymbol{y})$. The singleton c in $\boldsymbol{y}$ being clearly safe, we can remove it and continue as in the previous case. □

Remark 3.4 below providing a lower bound for the first 4 entries above the main diagonal of the $(d, n-d)$ table, is a corollary of the inequality $\rho_{d+s}(n+2s) \geq \rho_d(n) + s$, noted in [5], applied to $\rho_2(k) = k-3$ for $k = 5, 6, 7$ and 8.

Remark 3.4 *We have $\rho_{d-k}(2d-k) \geq d-1$ for $k = 1, 2, 3$ and 4 and $d \geq 6$.*

4 Structural properties of run-maximal strings on the main diagonal

We explore structural properties of the run-maximal strings on the main diagonal. These results yield properties for run-maximal strings that have their length bounded by nine times the number of distinct symbols they contain. We can thus shift the critical region of the $(d, n-d)$ table as summarized in the Theorem 4.1, the proof for which can be found at the end of this section.

Theorem 4.1 *We have $\{\rho_d(n) \leq n-d$ for $2 \leq d \leq n\} \Leftrightarrow \{\rho_d(9d) \leq 8d$ for $d \geq 2\}$.*

Proposition 4.2 describes useful structural properties of run-maximal strings on the main diagonal. The proof of the proposition relies on a few lemmas that will be mostly presented without their entire proofs, just a few examples will be given to illustrate the method. They all deal with the same basic scenario: assuming we know that the table obeys the conjecture for all columns to the left of column d, which is the first *unknown* column, we investigate the run-maximal strings of $S_d(2d)$.

Proposition 4.2 *Let $\rho_{d'}(2d') \leq d'$ for $2 \leq d' < d$. Let $\boldsymbol{x}$ be a run-maximal string in $S_d(2d)$. Either $r(\boldsymbol{x}) = \rho_d(2d) = d$ or $\boldsymbol{x}$ has at least $\lceil \frac{7d}{8} \rceil$ singletons, and no symbol occurs exactly 2, 3, ... 8 times in $\boldsymbol{x}$.*

Proof. The proof that each symbol must be a singleton or occur at least 9 times is a direct result of the lemmas which make up the remainder of this section. Then, let $\boldsymbol{x} \in S_d(2d)$ be run-maximal, m_1 denote the number of singletons, and m_2 the number of non-singleton symbols of $\boldsymbol{x}$. We have $m_1 + 9m_2 \leq 2d$ and $m_1 + m_2 = d$, which implies that $m_2 \leq d/8$ and hence $m_1 \geq \lceil 7d/8 \rceil$. □

Proposition 4.2 provides a purely structural proof that $\rho_d(2d) = d$ for $d \leq 15$, and using the computation of $\rho_2(d+2)$ for $d = 16, \ldots, 23$, that $\rho_d(2d) = d$ for $d \leq 23$.

Corollary 4.2.1 *We have $\rho_d(2d) = d$ for $d \leq 23$ and $\rho_d(n) \leq n - d$ for $n - d \leq 23$.*

Proof. Assume that run-maximal $\boldsymbol{x} \in S_d(2d)$ satisfies $r(\boldsymbol{x}) = \rho_d(2d) > d$. By Proposition 4.2, $\boldsymbol{x}$ consists only of singleton for $2 \leq d \leq 6$, $r(\boldsymbol{x}) = \rho_1(d+1) = 1$ for $8 \leq d \leq 15$, and $d < r(\boldsymbol{x}) = \rho_2(d+2)$ for $16 \leq d \leq 23$, which are impossible. □

In Lemmas 4.3,4.4, and 4.5 we assume that for $2 \leq d' < d$, the conjecture holds, i.e. $\rho_{d'}(2d') \leq d'$. Note that it is equivalent to $\rho_{d'}(n') \leq n' - d'$ for $2 \leq d' \leq n'$ when $n' - d' < d$. We consider a run-maximal string $\boldsymbol{x} \in S_d(2d)$ containing a k-tuple. We show that either the string $\boldsymbol{x}$ obeys the conjectured upper bound, or can be manipulated to obtain a new string $\boldsymbol{y}$ with a larger alphabet of the same or shorter length. We ensure that the manipulation process does not destroy more runs than the the amount the alphabet is increased or the length decreased. This allows us to estimate the number of runs in $\boldsymbol{y}$ based on the values in the table for some $d' < d$. In essence, we manipulate a string from column d to a string from some column $d' < d$ while monitoring the number of runs.

Lemma 4.3 *Let $\rho_{d'}(2d') \leq d'$ for $2 \leq d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) = d$ or $\boldsymbol{x}$ does not contain a pair.*

Proof. Assume that $\boldsymbol{x}$ does not obey the conjectured upper bound and so $r(\boldsymbol{x}) > d$. Let us assume that $\boldsymbol{x}$ contains a pair of C's and so $\boldsymbol{x} = \boldsymbol{u}C\boldsymbol{v}C\boldsymbol{w}$. Change the first occurrence of C to a new symbol $D \notin \mathcal{A}(\boldsymbol{x})$ to obtain $\boldsymbol{y} = \boldsymbol{u}D\boldsymbol{v}C\boldsymbol{w}$. Since a pair can be in at most one run (see for instance [5]), we destroyed at most one run and increased the alphabet size by one, so $d - 1 \geq \rho_{d+1}(2d) \geq r(\boldsymbol{y}) \geq r(\boldsymbol{x}) - 1$. It follows that $d \geq r(\boldsymbol{x})$, a contradiction with our earlier assumption. □

Lemma 4.4 *Let $\rho_{d'}(2d') \leq d'$ for $2 \leq d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) = d$ or $\boldsymbol{x}$ does not contain a triple.*

Proof. (A sketch) If $\boldsymbol{x}$ does not obey the conjecture and has a triple of C's, the triple can be involved in at most two runs. We change the first two occurrences of C to new symbols D and E obtaining $\boldsymbol{y} \in S_{d+2}(2d)$. This destroys at most two runs while increasing the size of the alphabet by 2, a contradiction with our assumption. □

For k-tuples of higher degree, $4 \leq k \leq 8$, the approach is very similar, but since such a k-tuple can be in multiple runs, the discussion of cases become more complex and thus we summarize all these results without a proof in Lemma 4.5.

Lemma 4.5 *Let $\rho_{d'}(2d') \leq d'$ for $2 \leq d' < d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. Either $r(\boldsymbol{x}) = \rho_d(2d) = d$ or $\boldsymbol{x}$ does not contain a k-tuple, $4 \leq k \leq 8$.*

While the previous lemmas were provided for entries on the main diagonal, the result can be generalized to any entry in column $n - d$ where $\rho_{d'}(n') \leq n' - d'$ for $n' - d' < n - d$. Either $\rho_d(n) \leq n - d$, or no run-maximal $\boldsymbol{x} \in S_d(n)$ has a pair, triple, ..., 8-tuple. The induction hypothesis only requires that all entries to the left of the *unknown* column satisfy the conjecture; there is no restriction within the *unknown* column.

Having proven Proposition 4.2, we can present the proof of Theorem 4.1: **Proof.** The proof follows directly from Proposition 4.2. If the conjecture does not hold, let d be the first column for which $\rho_d(2d) > d$. Let $\boldsymbol{x} \in S_d(2d)$ be run-maximal. By Proposition 4.2, $\boldsymbol{x}$ has at least $k = \lceil \frac{7d}{8} \rceil$ singletons, and by Lemma 2.2 they must all be safe. Let us form $\boldsymbol{y}$ by removing all these safe singletons. This gives a string $\boldsymbol{y} \in S_{d-k}(2d-k)$ violating the conjecture, i.e. $r(\boldsymbol{y}) > d$. $d' = d - k = \frac{d}{8}$ and $d = 8d'$ and $2d - k = 9d'$. Thus we found a $\boldsymbol{y} \in S_{d'}(9d')$ such that $r(\boldsymbol{y}) > 8d'$. □

When investigating a single column, the first counter-example in the column cannot have a singleton, as otherwise the counter-example could be *pushed up*. Nor, by Proposition 4.2, can it contain a k-tuple for $2 \leq k \leq 8$. Theorem 4.1 together with these facts give a simplified way to computationally *verify* that the whole column d satisfies the conjecture: *show that there are no counter-examples for* $2 \leq d' \leq \frac{d}{8}$*, and only strings with no* k*-tuples,* $1 \leq k \leq 8$*, need to be considered when looking for the counter-examples.*

5 Conclusion

The properties presented in this paper constrain the behaviour of the entries in the $(d, n-d)$ table below the main diagonal and in an immediate neighbourhood above the main diagonal. One of the the main contributions lies in the characterization of structural properties of the run-maximal strings on the main diagonal, giving yet another property equivalent with the maximum number of runs conjecture. Not only do these results provide a faster way to computationally check the validity of the conjecture for greater lengths, they indicate a possible way to prove the conjecture along the ideas presented in Proposition 4.2 and its proof: a first counter-example on the main diagonal could not possibly have a k-tuple for any conceivable k. We were able to carry the reasoning up to $k = 8$, but these proofs are not easy to scale up as the combinatorial complexity increases. The hope and motivation for further research along these lines is that there is a common thread among all these various proofs that may lead to a uniform method ruling out all the k-tuples and thus proving the conjecture, or to exhibit an unexpected counter-example on the main diagonal of the $(d, n-d)$ table.

References

[1] A. Baker, A. Deza, and F. Franek. Run-maximal strings. website, 2011. `http://optlab.mcmaster.ca/~bakerar2/research/runmax/`.

[2] M. Crochemore and L. Ilie. Maximal repetitions in strings. *Journal of Computer and System Sciences*, 74(5):796–807, 2008.

[3] M. Crochemore, L. Ilie, and L. Tinta. Towards a solution to the "runs" conjecture. *Lecture Notes in Computer Science*, 5029:290–302, 2008.

[4] M. Crochemore, L. Ilie, and L. Tinta. The "runs" conjecture. website, 2011. `http://www.csd.uwo.ca/faculty/ilie/runs.html`.

[5] A. Deza and F. Franek. A d-step analogue for runs on strings. AdvOL-Report 2010/02, McMaster University, 2010.

[6] K. Fan, S.J. Puglisi, W.F. Smyth, and A. Turpin. A new periodicity lemma. *SIAM Journal on Discrete Mathematics*, 20(3):656–668, 2006.

[7] K. Fan, W.F. Smyth, and R.J. Simpson. A new periodicity lemma. *Lecture Notes in Computer Science*, 3537:257–265, 2005.

[8] F. Franek and J. Holub. A different proof of Crochemore-Ilie lemma concerning microruns. In *London Algorithmics 2008: Theory and Practice*, pages 1–9. College Publications, London, UK, 2009.

[9] F. Franek, R.J. Simpson, and W. Smyth. The maximum number of runs in a string. In *Proceedings of 14th Australasian Workshop on Combinatorial Algorithms AWOCA 2003*. Seoul National University, Seoul, Korea, 2008.

[10] F. Franek and Q. Yang. An asymptotic lower bound for the maximal number of runs in a string. *International Journal of Foundations of Computer Science*, 19(1):195–203, 2008.

[11] M. Giraud. Not so many runs in strings. In *LATA 2008*. Tarragona, Spain, 2008.

[12] C.S. Iliopoulos, D. Moore, and W.F. Smyth. A characterization of the squares in a Fibonacci string. *Theoretical Computer Science*, 172:281–291, 1997.

[13] R. Kolpakov and G. Kucherov. Finding maximal repetitions in a word in linear time. In *40th Annual Symposium on Foundations of Computer Science*, pages 596–604, 1999.

[14] R. Kolpakov and G. Kucherov. On maximal repetitions in words. In *Proc. 12th Intl. Symp. on Fund. of Comp. Sci. 1999*, volume 1684, pages 374–385, 1999.

[15] E. Kopylov and W.F. Smyth. The three squares lemma revisited. to appear.

[16] M.G. Main. Detecting leftmost maximal periodicities. *Discrete Applied Mathematics*, 25:145–153, 1989.

[17] W. Matsubara, K. Kusano, H. Bannai, and A. Shinohara. A series of run-rich strings. *Lecture Notes in Computer Science*, 5457:578–587, 2009.

[18] W. Matsubara, K. Kusano, A. Ishino, H. Bannai, and A. Shinohara. New lower bounds for the maximum number of runs in a string. In *Proceedings of PSC 2008*, pages 140–145. Czech Technical University, Prague, Czech Republic, 2008.

[19] W. Matsubara, K. Kusano, A. Ishino, H. Bannai, and A. Shinohara. Lower bounds for the maximum number of runs in a string. website, 2011. `http://www.shino.ecei.tohoku.ac.jp/runs/`.

[20] S.J. Puglisi, R.J. Simpson, and W.F. Smyth. How many runs can a string contain? *Theoretical Computer Science*, 401:165–171, 2008.

[21] S.J. Puglisi, W.F. Smyth, and A. Turpin. Some restrictions on periodicity in strings. In *Proceedings of the 16th Australasian Workshop on Combinatorial Algorithms*, pages 415–428, 2005.

[22] W. Rytter. The number of runs in a string: Improved analysis of the linear upper bound. *Lecture Notes in Computer Science*, 3884:184–195, 2006.

Andrew Baker, Antoine Deza, and Frantisek Franek
Advanced Optimization Laboratory,
Department of Computing and Software,
McMaster University, Hamilton, Ontario, Canada.
Email: bakerar2, deza, franek@mcmaster.ca