On the Average Complexity for the Verification of Compatible Sequences

Christos Koukouvinos[a], Veronika Pillwein[b], Dimitris E. Simos[a,*], Zafeirakis Zafeirakopoulos[b]

[a] *National Technical University of Athens, Zografou 15773, Athens, Greece*
[b] *Research Institute for Symbolic Computation (RISC), Linz, Austria*

Abstract

The average complexity analysis for a formalism pertaining pairs of compatible sequences is presented. The analysis is done in two levels, so that an accurate estimate is achieved. The way of separating the candidate pairs into suitable classes of ternary sequences is interesting, allowing the use of fundamental tools of symbolic computation, such as holonomic functions and asymptotic analysis to derive an average complexity of $O(n\sqrt{n}\log n)$ for sequences of length n.

Keywords: Sequences, Periodic Autocorrelation Function, Non-Periodic Autocorrelation Function, Complexity, Algorithms, Average-case analysis, Symbolic Computation
2000 MSC: 05B20, 68Q25, 68W40, 33F10, 68W30

1. Introduction

In this paper, we detail an average complexity analysis for a formalism that exhibits the cross-fertilization of combinatorics with theoretical computer science, in the study of sequences with constant (non) periodic autocorrelation function.

In what follows we present the asymptotic analysis for the average case complexity of deciding if two sequences have PAF/NPAF (see Definition 1) equal to a constant α. We employ a fine grained analysis initially and then we provide the asymptotics for a pair of sequences of length n. In the course of the analysis we extensively use tools from Computer Algebra and especially tools for dealing with holonomic functions and their asymptotics since such functions occur in the summations needed for the analysis.

In Section 2 we provide the necessary definitions and previous work. In Section 3 we describe the algorithms and the essential part of the theory behind them. In Section 4 we detail the asymptotic analysis for the average case complexity of the algorithms. In Section 5 we conclude by giving some hints on the practical complexity and arguing on the efficiency of the algorithms examined.

2. Preliminaries

This section gives a collection of notions that are used throughout the paper.

Definition 1. *For a sequence $A = [a_1, a_2, \ldots, a_n]$ of length n the periodic autocorrelation function (PAF) and the non-periodic autocorrelation function (NPAF), denoted by $P_A(s)$ and $N_A(s)$, are defined as*

$$P_A(s) = \sum_{i=1}^{n} a_i a_{i+s}\ , s = 0, 1, \ldots, n-1, \qquad \text{and} \qquad N_A(s) = \sum_{i=1}^{n-s} a_i a_{i+s}\ , s = 0, 1, \ldots, n-1,$$

respectively, where in PAF we consider (i+s) modulo n, see also [11, 13].

Definition 2. *Two sequences, $A = [a_1, \ldots, a_n]$ and $B = [b_1, \ldots, b_n]$, of length n are said to have PAF (respectively NPAF) equal to α, if $P_A(s)+P_B(s) = \alpha$ (respectively $N_A(s)+N_B(s) = \alpha$) for $s = 1, \ldots, n-1$.*

*Corresponding author

Following [5] the sequences A and B will be called compatible, if α is a constant. Note that such pairs of sequences are said to have constant (non) periodic autocorrelation function even though it is the sum of their autocorrelations that is a constant.
We are interested in bundling together the indices of entries with the same sign. This motivates the following definitions:

Definition 3. *The positive and negative support of a sequence $A = [a_1, \ldots, a_n]$, denoted by $POS(A)$ and $NEG(A)$, respectively, are defined as*

$$POS(A) = \{i : a_i > 0,\ i = 1, \ldots, n\} \qquad \text{and} \qquad NEG(A) = \{j : a_j < 0,\ j = 1, \ldots, n\},$$

while its weight $w(A)$ is defined as $w(A) = |POS(A)| + |NEG(A)|$.

Definition 4. *We define the occurrences counting function $[S]_e$ for a multiset S and an element from the domain of elements of S as $[S]_e = |[x \in S \ : \ x = e]|$.*

For example, let S be the multiset $S = [1, 1, 2, 2, 2, 4]$. Then $[S]_1 = 2, [S]_2 = 3, [S]_3 = 0$ and $[S]_4 = 1$.

Lemma 1. *Let A, B be multisets and by $A \uplus B$ denote the union of A and B, preserving all multiplicities. Then $[A \uplus B]_e = [A]_e + [B]_e$.*

For prior usage of multisets in the study of sequences with constant (non) periodic autocorrelation function we refer to [7, 15, 20], while for related operations on them see [12].

3. A Combinatorial Algorithm for Compatible Sequences

In this section, we present an algorithm that based on their support decides if two sequences are compatible. The use of the support of sequences first appeared in [5] and [6], and recently for such computations in [7] and [15].

3.1. Sets of Differences

In order to state the algorithm, we need to define multisets of differences on their support.

Definition 5. *We define the following multisets:*

- *Signed differences in the positive support of A: $D_A^+ = [x - y \ : \ x, y \in POS(A), x > y]$.*
- *Signed differences in the negative support of A: $D_A^- = [x - y \ : \ x, y \in NEG(A), x > y]$.*
- *Cross differences between the positive and negative support of A: $C_A = D_A^{\pm} \uplus D_A^{\mp}$, where $D_A^{\pm} = [x - y \ : \ x \in POS(A), y \in NEG(A), x > y]$ and $D_A^{\mp} = [x - y \ : \ x \in NEG(A), y \in POS(A), x > y]$.*

Remark 1. *Depending on whether we are interested in PAF or NPAF:*

- *in the definition of the sets above, for PAF we use $(x - y) \pmod n$ instead of $(x - y)$*
- *the notation $AF_A(s)$ denotes either $PAF_A(s)$ or $NPAF_A(s)$*

Notation 1. *We denote by $AF_{A,B}(s)$ the sum $AF_A(s) + AF_B(s)$. If the sequences A and B are clear from the context, we denote $AF_{A,B}(s)$ by $AF(s)$.*

The motivation behind the multisets in Definition 5 is counting the contribution of each sequence in $AF_{A,B}(s)$.

Lemma 2. *Let A, B be two sequences of length n, weight w and entries from $\{-1, 0, 1\}$. Let D be $D_A^+ \uplus D_A^- \uplus D_B^+ \uplus D_B^-$ and C be $C_A \uplus C_B$. For $s \in \{1, 2, \ldots n-1\}$, the following are equivalent:*

(i) $AF_{A,B}(s) = \alpha$

(ii) $[D]_s - [C]_s = \alpha$

PROOF. Fix $s \in \{1, 2, \ldots n-1\}$. We prove the lemma for NPAF. The PAF case is similar. We have that $AF_{A,B}(s) = AF_A(s) + AF_B(s)$. Thus we need to compute $AF_C(s)$ for $C = A, B$. We have that $AF_C(s) = |P_C| - |N_C|$, where $P_C = [(c_i, c_{i+s}) \ : \ c_i c_{i+s} = 1, \ i = 1, 2 \ldots, n-s, \ c_i \in C]$ and $N_C = [(c_i, c_{i+s}) \ : \ c_i c_{i+s} = -1, \ i = 1, 2 \ldots, n-s, \ c_i \in C]$.

We should consider only elements of the support, since all products involving a zero element contribute zero to the sum. The pairs of elements of the support contributing a "+1" are the ones where both elements have the same sign, i.e., elements of P_C. By the definition of D_C^+ (respectively D_C^-), for every occurrence of s in D_C^+ (respectively D_C^-), there is a pair (c_i, c_{i+s}) for some i such that c_i and c_{i+s} have both positive (respectively negative) sign, thus belonging to P_C. The other direction, i.e., for each pair $(c_i, c_{i+s}) \in P_C$ there is an occurrence of s in D_C^+ or D_C^- follows directly by the definition of D_C^+ and D_C^-. Thus the cardinality of P_C is equal to the number of occurrences of s in D_C^+ and D_C^-.

Similarly, the pairs of elements of the support contributing a "−1" are the ones where elements have opposite signs. By the definition of C_C, for every occurrence of s in C_C, there is a pair (c_i, c_{i+s}) for some i such that c_i and c_{i+s} have opposite signs, thus belonging to N_C. The other direction, i.e., for each pair $(c_i, c_{i+s}) \in N_C$ there is an occurrence of s in C_C follows directly by the definition of C_C. Thus the cardinality of N_C is equal to the number of occurrences of s in C_C.

Summarizing for A and B we have

$$\begin{aligned} AF_{A,B}(s) &= AF_A(s) + AF_B(s) \\ &= |P_A| - |N_A| + |P_B| - |N_B| \\ &= (|P_A| + |P_B|) - (|N_A| + |N_B|) \\ &= ([D_A^+]_s + [D_A^-]_s + [D_B^+]_s + [D_B^-]_s) - ([C_A]_s + [C_B]_s) \\ &= [D_A^+ \uplus D_A^- \uplus D_B^+ \uplus D_B^-]_s - [C_A \uplus C_B]_s \\ &= [D]_s - [C]_s \end{aligned}$$

Thus

$$AF_{A,B}(s) = \alpha \Leftrightarrow [D]_s - [C]_s = \alpha$$

□

Depending on the value of α and the choice of $AF(s)$ we can now verify the AF property for a number of combinatorial objects. For example, a ternary complementary pair (TCP) made up of two sequences A and B is defined as $TCP(n, w) := \{(A, B) : A, B \in \{-1, 0, 1\}^n, \ w(A) + w(B) = w, \ NPAF_{A,B}(s) = 0, \ s = 1, \ldots, n-1\}$ (see [3]). Then we can simply check for $NPAF_{A,B}(s) = 0 \Leftrightarrow [D]_s = [C]_s$. Note that in this case the derived multisets must be equal. We give another example based on generalized Legendre pairs (GL-pairs) of length ℓ defined as $GL(\ell) := \{(A, B) : A, B \in \{-1, 1\}^\ell, \ w(A) + w(B) = 2\ell, \ PAF_{A,B}(s) = -2, \ s = 1, \ldots, n-1\}$ (see [5]). Similarly we check for $PAF_{A,B}(s) = -2 \Leftrightarrow [C]_s - [D]_s = 2$. In the sequel, GL-pairs and TCPs are used to produce Hadamard and weighing matrices (see Theorem 3 of [5] and Theorem 5 of [14]). Design Theory is rich of sequences with similar properties [11, 13, 14], with numerous applications in signal processing [8, 9] and Cryptography [18].

3.2. Description of the Algorithm

From Lemma 2 we can deduce an algorithm for the verification of the following property,

"The sequences A and B of length n have constant (non) periodic autocorrelation equal to α."

The main steps of the algorithm are:

- Compute the support
- Compute the multisets of signed differences and cross signed differences
- Verify that $[D]_s - [C]_s = \alpha$ for $s \in \{1, 2, \ldots, n-1\}$

These steps extend the results of [15] in the case of $\alpha \neq 0$, and those of [7] for two ternary sequences with zero AF. We give a high level description of algorithm 1, below.

Algorithm 1 AF Computation Algorithm

procedure AFVERIFICATION(A, B, α)
Require: A, B are $\{0, \pm 1\}$ sequences of length n and $\alpha \in \{0, 1, \ldots, 2n\}$
  $n \leftarrow |A|$
  Compute $POS(A), NEG(A), POS(B), NEG(B)$ as in Definition 3
  Compute $D_A^+, D_A^-, C_A, D_B^+, D_B^-, C_B$ as in Definition 5
  Compute $D = D_A^+ \uplus D_A^- \uplus D_B^+ \uplus D_B^-$ and $C = C_A \uplus C_B$
  for $s = 1, 2, \ldots, n-1$ **do**
    if $[D]_s - [C]_s \neq \alpha$ **then return** False
    end if
  end for
return True
end procedure

4. Average Case Complexity Analysis

In this section we examine the average case complexity of Algorithm 1. Average-case analysis is a hard task to accomplish because there are a lot of details involved. First the different groups into which all possible input can be grouped need to be determined. The second step is to determine for each of the groups the probability for a random input to come from this group, under the assumption that every input is equally likely. The third step is to determine for each of the groups the complexity of the algorithm for input coming from this group. Finally, the average case complexity is given by the following formula:

$$AC(n) = \sum_{i=1}^{m} P_i T_i \tag{1}$$

where n is the size of the input, m is the number of groups, P_i is the probability assigned to the i^{th} group, and T_i is the complexity of the algorithms for input from the i^{th} group. We note that m, P_i, T_i (may) depend on n.

In what follows, we consider as input a pair of sequences. The total length is $2n$, w_1 is the weight of the first sequence, w_2 is the weight of the second sequence and $w = w_1 + w_2$.

Remark 2. *The following analysis is in the arithmetic model, i.e., all arithmetic operations are exact and contribute the same to the complexity of the algorithm.*

As a side note we should mention that, concerning bit-size complexity, the bit-size of any quantity appearing in the algorithm is bounded by $\log(n)$.

The following sets, along with the respective cardinalities are needed:

- Let $S(n) = \{(A, B) \in (\{0, \pm 1\}^n)^2\}$, then

$$|S(n)| = 3^{2n}, \tag{2}$$

  since there are 3 choices for each position in two sequences of length n.

- Let $S_w(n) = \{(A, B) \in S(n) \ : \ w(A) + w(B) = w\}$, then

$$|S(n)| = 2^w \binom{2n}{w}, \tag{3}$$

  since we choose w positions in two sequences of length n and we have 2 choices for each position.

- Let $S_{w_1,w_2}(n) = \{(A,B) \in S(n) \ : \ w(A) = w_1, w(B) = w_2\}$, then

$$|S_{w_1,w_2}(n)| = \binom{n}{w_1}\binom{n}{w_2}2^{w_1+w_2}, \tag{4}$$

since there are 2 choices (positive or negative) for each non zero entry, $\binom{n}{w_1}$ choices for the non zero entries in the first sequence and $\binom{n}{w_2}$ choices for the non zero entries in the second sequence.

4.1. Complexity

We now examine the complexity of the algorithm for an input from $S^{a,b}_{w_1,w_2}(n)$. The algorithm consists of three steps as explained previously.

- Step 1: Compute the support
  This step requires $4n$ operations in order to scan the sequences and populate the sets POS(A),POS(B), NEG(A) and NEG(B).

- Step 2: Compute the signed difference sets
  This step requires at most 3 operations for each element added in any of the multisets D and C, where $D = D_A^+ \uplus D_A^- \uplus D_B^+ \uplus D_B^-$ and $C = C_A \uplus C_B$. To obtain the number operations we count how many elements each of the sets involved has:
  - $|D_A^+| = \frac{a(a-1)}{2}$ for $a = |POS(A)|$
  - $|D_A^-| = \frac{\bar{a}(\bar{a}-1)}{2}$ for $\bar{a} = |NEG(A)|$
  - $|D_B^+| = \frac{b(b-1)}{2}$ for $b = |POS(B)|$
  - $|D_B^-| = \frac{\bar{b}(\bar{b}-1)}{2}$ for $\bar{b} = |NEG(B)|$
  - $|C_A| = a\bar{a}$ for $a = |POS(A)|$, $\bar{a} = |NEG(A)|$
  - $|C_B| = b\bar{b}$ for $b = |POS(B)|$, $\bar{b} = |NEG(B)|$

  Thus, $|D| = \frac{1}{2}(a(a-1) + \bar{a}(\bar{a}-1) + b(b-1) + \bar{b}(\bar{b}-1))$ and $|C| = a\bar{a} + b\bar{b}$.
  Using $\bar{a} = w_1 - a$ and $\bar{b} = w_2 - b$ yields that the complexity of this step is $\frac{1}{2}(w_1^2 - w_1 + w_2^2 - w_2)$.

- Verification of $AF = \alpha$
  1. $D \log D + C \log C$ steps to sort the two multisets D and C.
  2. $\max\{D, C\}$ steps are needed to decide if $[D]_s - [C]_s = \alpha$ for all $s \in D \cup C$, scanning through the sorted multisets.

  The scanning is dominated by the sorting, thus this step contributes $a\bar{a}\log(a\bar{a}) + b\bar{b}\log(b\bar{b})$ to the complexity. In order to simplify the computations we will bound this quantity by something not depending on a and b.

  For this particular case it is easy to compute such a bound (by computing the maximum of a quadratic equation). Nevertheless, we use Cylindrical Algebraic Decomposition (cf. [2, 19]), which is a standard and powerful tool for this task (substituting $\bar{a}$ by $w_1 - a$ and $\bar{b}$ by $w_2 - b$):

```
In[1]:= Resolve[ForAll[{a}, 0 <= a <= w1, a(w1 - a) <= M], {w1, M}, Reals]

Out[1]= w1 > 0 && M >= w1^2/4
```

  Thus we can bound $a(w_1 - a) \leq \frac{w_1^2}{4}$ and $b(w_2 - b) \leq \frac{w_2^2}{4}$ and as a consequence the costs for the verification of $AF = \alpha$ can be bounded by

$$\frac{w_1^2}{4}\log\frac{w_1^2}{4} + \frac{w_2^2}{4}\log\frac{w_2^2}{4} \leq \frac{w_1^2}{2}\log w_1 + \frac{w_2^2}{2}\log w_2 \leq \tfrac{1}{2}(w_1^2 + w_2^2)\log(w_1 + w_2) \leq \tfrac{1}{2}(w_1^2 + w_2^2)\log n.$$

Adding the complexities of each step yields the upper bound

$$\frac{1}{2}\left(8n - w_1 + w_1^2 - w_2 + w_2^2 + w_1^2 \log n + w_2^2 \log n\right).$$

Since we are only interested in a big-Oh estimate of the complexity, we can safely ignore constant factors and simplify further by dropping the terms that are subtracted and bounding w_i^2 by $w_i^2 \log n$ from above for $i = 1, 2$. Thus we conclude that for input from $S_{w_1,w_2}(n)$ the complexity of the algorithm is of order

$$\mathcal{O}\left(n + \log n \left(w_1^2 + w_2^2\right)\right). \tag{5}$$

4.2. Probabilities

For the average case analysis we need the probability for a pair of sequences of length n to have weights w_1 and w_2:

$$P_1(w_1, w_2, n) = \frac{|S_{w_1,w_2}(n)|}{|S(n)|} = \frac{2^{w_1+w_2}\binom{n}{w_1}\binom{n}{w_2}}{3^{2n}} \tag{6}$$

and the probability for a pair of sequences of length n to have total weight w:

$$P_2(w, n) = \frac{|S_w(n)|}{|S(n)|} = \frac{2^w\binom{2n}{w}}{3^{2n}} \tag{7}$$

4.3. Average Complexity

We will repeat the process of finding the average complexity two times, at different level of detail.

At the first level we examine the complexity considering input from $S_w(n)$ and obtain the average case complexity $C_2(w, n)$ separating the possible $S_w(n)$ into groups according to w_1, w_2 (i.e., $S_{w_1,w_2}(n)$). Since we average over the possible weights for a given length, we consider n to be fixed and we can ignore the linear term n and the $\log n$ factor of the second term in (5). In the final step we will reintroduce these terms to obtain the average case complexity. Let

$$C_1(w_1, w_2) = w_1^2 + w_2^2 \tag{8}$$

then according to (1):

$$C_2(w, n) = \sum_{w_1+w_2=w} P_1(w_1, w_2, n) C_1(w_1, w_2) = \sum_{i=0}^{w} P_1(i, w-i, n) C_1(i, w-i) \tag{9}$$

At the second level we average over the weight, and we compute the average case complexity for input from $S(n)$. According to (1) again, we have:

$$C_3(n) = \sum_{w=0}^{2n} P_2(w, n) C_2(w, n) \tag{10}$$

Then the average complexity is

$$AC(n) = \mathcal{O}(n + C_3(n) \log(n)) \tag{11}$$

4.4. Computations

In this section, we carry out the computations for the aforementioned analysis, using MATHEMATICA. From (8) and (9), we have

$$\begin{aligned} C_2(w, n) &= \sum_{i=0}^{w} \frac{2^{i+w-i}\binom{n}{i}\binom{n}{w-i}}{3^{2n}} \left(i^2 + (w-i)^2\right) \\ &= \frac{2^{w+1}}{3^{2n}} \sum_{i=0}^{w} \binom{n}{i}\binom{n}{w-i} i^2 \\ &= \frac{2^w}{3^{2n}} \frac{w(n - w - nw)}{2n-1} \binom{2n}{w}. \end{aligned} \tag{12}$$

Combining (10) and (12) yields

$$C_3(n) = \sum_{w=0}^{2n} P_2(w,n)C_2(w,n) = \sum_{w=0}^{2n} \frac{2^w\binom{2n}{w}}{3^{2n}} \frac{2^w}{3^{2n}} \frac{w(n-w+nw)}{2n-1}\binom{2n}{w}$$
$$= 3^{-4n}\frac{1}{2n-1}\sum_{w=0}^{2n} 2^{2w}\binom{2n}{w}^2 w(n-w+nw).$$

The expression for the average complexity $C_3(n)$ is a very well behaved object. It satisfies a linear recurrence in n with polynomial coefficients and as such it fits into the so-called holonomic framework [22, 1, 16]. Within the last decades several symbolic algorithms have been designed and implemented that are capable of automatically finding (and thus proving!) recurrence relations for this type of input [17]. We are interested in the asymptotic behavior of the average complexity and to obtain this information we first compute a recurrence relation for the sum in (4.4). From this recurrence relation then the asymptotic behavior can easily be determined using standard techniques, see e.g. [4, 21]. For the computation of the recurrence relations we employ the MATHEMATICA package HolonomicFunctions[1] implemented by Christoph Koutschan [16]. The package is loaded in the MATHEMATICA kernel via

```
In[2]:= << HolonomicFunctions.m
        HolonomicFunctions package by Christoph Koutschan, RISC-Linz, Version 0.13 (13.05.2009) —
        Type ?HolonomicFunctions for help
```

To obtain a recurrence for the terms in (4.4) we first compute the annihilator with respect to shifts in n, where the shift operator is denoted by S_n.

```
In[3]:= ann = [Annihilator[3^(-4n) 1/(2n-1) Sum[2^(2w) Binomial[2n,w]^2 w(n-w+nw), {w,0,2n}], {S[n]}]]

Out[3]= {81n(n+1)^2(2n+3)(4n+1)(40n^2+20n-39)S_n^2 + (n+1)^2(n+2)(2n-1)(4n+5)(40n^2+100n+21)
         -2n(n+2)(4n+3)(3280n^4+9840n^3+2722n^2-6987n-1781)S_n}
```

The final step consists in determining the asymptotic behavior of the recurrence. For this purpose we make use of a MATHEMATICA-implementation of the methods described in [21] by Manuel Kauers [10] in the package Asymptotics[1]. The input is merely the recurrence relation satisfied by the given sequence, i.e.,

```
In[4]:= Asymptotics[81n(n+1)^2(2n+3)(4n+1)(40n^2+20n-39) f[n+2]
        - 2n(n+2)(4n+3)(3280n^4+9840n^3+2722n^2-6987n-1781) f[n+1]
        + (n+1)^2(n+2)(2n-1)(4n+5)(40n^2+100n+21) f[n], f[n]]

Out[4]= {n^(3/2)/81^n, n^(3/2)}
```

The output describes the asymptotic behavior of the given sequence, i.e., in our case it is $n^{3/2}\left(c_1 + c_2 3^{-4n}\right)$. Going back to (11) summarizing we have that the average complexity of the algorithm is:

$$AC(n) = \mathcal{O}(n + n^{\frac{3}{2}}\log(n)) = \mathcal{O}(n\sqrt{n}\log n) \tag{13}$$

It is possible to determine an approximation to the constant c_1 of the dominant term $n^{3/2}$ in the asymptotic behavior. Following the procedure outlined in [10] we can apply again the Asymptotics package to determine further terms in the expansion of $C_3(n)$ and by considering the ratio of this expansion and $n^{3/2}$ thus obtain an approximation of c_1. The value

$$c_1 \approx 2.6586807763582740409472512250117177741963241841836$$

[1] available at `http://www.risc.jku.at/research/combinat/software/`

is the approximation if we consider up to 50 digits in an expansion of order 15. For identifying which real number this approximation corresponds to we employ the online-available inverse symbolic calculator [2], which gives striking evidence that $c_1 = \frac{3}{2}\sqrt{\pi}$.

Figure 4.4 shows the graph of $c_1 n\sqrt{n}\log n$ and the graph of the real performance of a (naive) implementation on random data. The x-axis is the length n and the y-axis is the time. For each n we used 50 random pairs of sequences of length n.

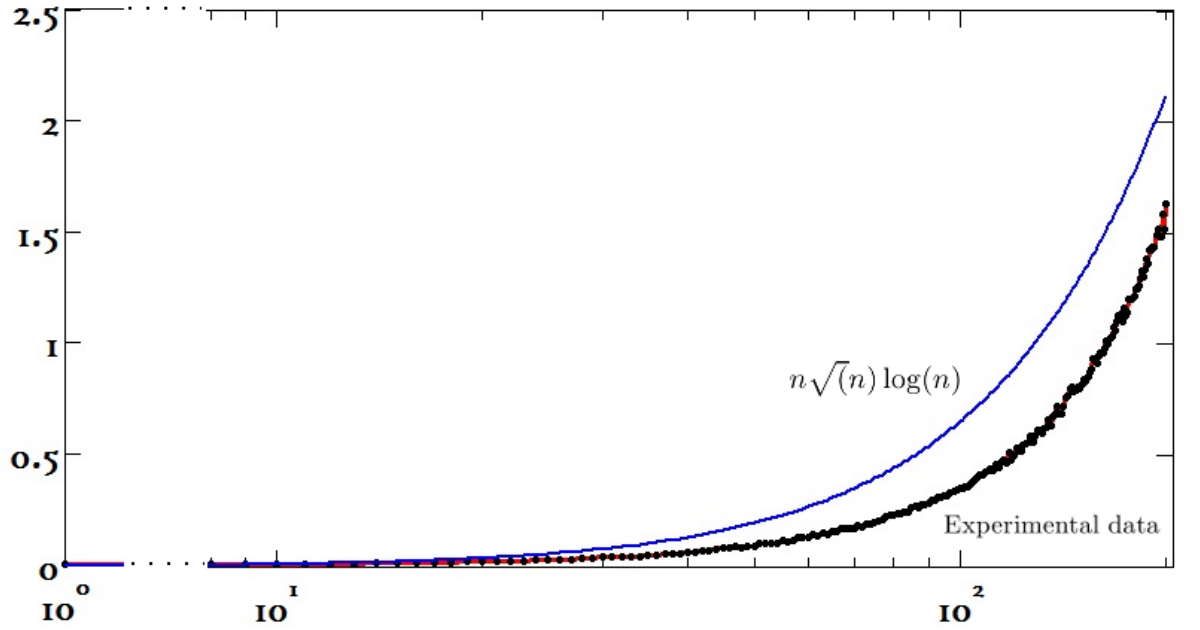

Figure 1: Comparison of average case complexity against real performance.

The experimental data confirms our analysis.

5. Conclusion

We presented a detailed analysis for the average case complexity of an algorithm deciding if two sequences are compatible based on their support. In the course of the analysis we have used different tools from computer algebra and computed the average case asymptotic complexity to be $\mathcal{O}(n\sqrt{n}\log n)$. This estimate proves that in practice the algorithm under consideration performs better than what the worst case complexity ($\mathcal{O}(n^2 \log n)$, see [15]) suggests.

Acknowledgements

The authors are thankful to the referees for their comments and suggestions that lead to an improvement in the quality and the presentation of the paper. The authors are grateful to Manuel Kauers for providing the asymptotics package. The second and fourth author are supported by the Austrian Science Fund (FWF) grant W1214/DK6. Part of this work was completed while the third author was visiting RISC in Linz. Thanks go to the RISC for the kind hospitality and support from the RISC Transnational Access Programme supported by the European Commission FP6 for Integrated Infrastructures Initiatives under the project SCIEnce.

[2] `http://oldweb.cecm.sfu.ca/projects/ISC/`

References

[1] F. Chyzak, An extension of Zeilberger's fast algorithm to general holonomic functions, Discrete Math. 217 (2000) 115–134.

[2] G. Collins, Quantifier elimination for real closed fields by cylindrical algebraic decompostion, in: H. Brakhage (Ed.), Automata Theory and Formal Languages 2nd GI Conference Kaiserslautern, May 20–23, 1975, volume 33 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 1975, pp. 134–183.

[3] R. Craigen, C. Koukouvinos, A theory of ternary complementary pairs, J. Combin. Theory Ser. A 96 (2001) 358–375.

[4] P. Flajolet, R. Sedgewick, Analytic combinatorics, Cambridge, Cambridge University Press, 2009.

[5] R.J. Fletcher, M. Gysin, J. Seberry, Application of the discrete fourier transform to the search for generalized legendre pairs and hadamard matrices, Australas. J. Combin. 23 (2001) 75–86.

[6] S. Georgiou, C. Koukouvinos, New infinite classes of weighing matrices, Sankhyā 64-B (2002) 26–36.

[7] S.D. Georgiou, Signed differences for weighing designs, Sankhyā 72-B (2010) 107–121.

[8] S. Golomb, H. Taylor, Two-dimensional synchronization patterns for minimum ambiguity, IEEE Trans. Inform. Theory 28 (1982) 600–604.

[9] J. Hersheya, R. Yarlagadda, Two-dimensional synchronisation, Electron. Lett. 19 (1983) 801–803.

[10] M. Kauers, A Mathematica Package for Computing Asymptotic Expansions of Solutions of P-Finite Recurrence Equations, Technical Report 00-00, RISC Report Series, University of Linz, Austria, 2011.

[11] H. Kharaghani, C. Koukouvinos, Complementary, base and turyn sequences, in: C.J. Colbourn, J.H. Dinitz (Eds.), Handbook of Combinatorial Designs, CRC Press, Boca Raton, Fla., 2006, 2nd edition, pp. 317–321.

[12] D.E. Knuth, The Art of Computer Programming, volume 2: Seminumerical Algorithms of *Addison-Wesley Series in Computer Science and Information Processing*, Addison-Wesley Publishing Co., Mass.- London-Don Mills, 3rd edition, 1998.

[13] C. Koukouvinos, Sequences with zero autocorrelation, in: C.J. Colbourn, J.H. Dinitz (Eds.), The CRC Handbook of Combinatorial Designs, CRC Press, Boca Raton, Fla., 1996, pp. 452–456.

[14] C. Koukouvinos, J. Seberry, New weighing matrices and orthogonal designs constructed using two sequences with zero autocorrelation function-a review, J. Statist. Plann. Inference 81 (1999) 153–182.

[15] C. Koukouvinos, D.E. Simos, On the computation of the non-periodic autocorrelation function of two ternary sequences and its related complexity analysis, J. Appl. Math. & Informatics 29 (2011) 547–562.

[16] C. Koutschan, HolonomicFunctions (User's Guide), Technical Report 10-01, RISC Report Series, University of Linz, Austria, 2010.

[17] H.W. M. Petkovšek, D. Zeilberger, $A = B$, A K Peters Ltd., Wellesley, MA, 1996.

[18] B. Schneier, Applied Cryptography, John Wiley and Sons, New York, 2nd edition, 1996.

[19] A. Strzeboński, Solving systems of strict polynomial inequalities, Journal of Symbolic Computation 29 (2000) 471–480.

[20] J.S. Wallis, On supplementary difference sets, Aequationes Math. 8 (1972) 242–257.

[21] J. Wimp, D. Zeilberger, Resurrecting the asymptotics of linear recurrences, J. Math. Anal. Appl. 111 (1985) 162–176.

[22] D. Zeilberger, A holonomic systems approach to special functions identities, J. Comput. Appl. Math. 32 (1990) 321–368.