IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 20, 2021, accepted April 28, 2021, date of publication May 4, 2021, date of current version May 12, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3077541

Low Autocorrelation Binary Sequences: Best-Known Peak Sidelobe Level Values

JANEZ BREST, (Senior Member, IEEE), AND BORKO BOŠKOVIĆ, (Member, IEEE)
Faculty of Electrical Engineering and Computer Science, Institute of Computer Science, University of Maribor, 2000 Maribor, Slovenia
Corresponding author: Janez Brest (janez.brest@um.si)

This work was supported by the Slovenian Research Agency (Computer Systems, Methodologies, and Intelligent Services) under Grant P2-0041.

ABSTRACT Binary sequences are widely used in many practical fields, such as radar applications, telecommunications and cryptography. Finding low autocorrelation binary sequences with good peak side-lobe level (PSL) values is a difficult optimization problem. In this paper we present an improved heuristic algorithm for searching low autocorrelation PSL sequences. A heuristic algorithm can find a sequence with a PSL value, which is not necessarily optimal, but is usually near optimal, and the algorithm finds it in a reasonable amount of time. In the experimental work we applied our algorithm to find binary sequences with low PSL values, and made a comparison with the state-of-the-art algorithms from literature. With our algorithm many sequences with the currently best-known PSL values have been improved. We found new sequences with better, i.e., lower, PSL values.

INDEX TERMS Binary code, aperiodic autocorrelation, peak sidelobe level.

I. INTRODUCTION

Low autocorrelation binary sequences (LABS) play important roles in many areas, such as communication engineering, synchronization, active sensing systems, cryptography and radar applications [1]–[5]. Searching for LABS with the lowest-achievable PSL values is a challenging optimization problem.

Generally, we have to distinguish between aperiodic and periodic sequences (codes). In this paper, aperiodic binary sequences are considered.

A binary sequence $S = s_1\, s_2 \ldots s_L$ has all entries either $+1$ or -1. Here, L denotes the sequence length. The *aperiodic autocorrelation function* (AACF) of binary sequence S at shift k is defined as:

$$C_k(S) = \sum_{i=1}^{L-k} s_i s_{i+k}, \text{ for } k = 0, \pm 1, \ldots, \pm(L-1). \quad (1)$$

Note that the AACF is an even function, since $C_k(S) = C_{(-k)}(S)$, and therefore, it is enough to consider it for the interval $k = 0, 1, \ldots, (L-1)$ only. The *Peak Sidelobe Level* (PSL) is the measure of smallness of the aperiodic autocorrelations and the PSL value is defined as:

$$\mathrm{PSL}(S) = \max_{1<k<L} |C_k(S)|. \quad (2)$$

The $C_0(S)$ is called the *mainlobe* level, and this term is not included in Eq. (2). The rest, $C_k(S)$, $k = 1, 2, \ldots, L-1$, are called *sidelobe* levels. The PSL value represented in decibels is given as:

$$\mathrm{NPSL}(S)\,[\mathrm{dB}] = 20 \log_{10}\Big(\frac{\mathrm{PSL}(S)}{L}\Big). \quad (3)$$

The LABS problem involves assigning values to the s_i that minimize PSL(S) values for all possible binary sequences of length L.

The search space of the LABS problem is of size 2^L. To locate good (optimal) solutions, two approaches exist: *Complete* and *incomplete* search. The complete, or exact search, is able to find the optimal sequence, but it is unlikely to scale up to large sequences. The incomplete, or stochastic search, can obtain a result that may be optimal or close to the optimal, i.e., it does not guarantee optimality.

Many authors have put considerable computational effort into finding binary sequences with small peak sidelobe level [6], [7], showing that:

- PSL(L) ≤ 2 for $L \leq 21$,
- PSL(L) ≤ 3 for $L \leq 48$,

The associate editor coordinating the review of this manuscript and approving it for publication was Xiaokang Wang.

- PSL$(L) \leq 4$ for $L \leq 82$,
- PSL$(L) \leq 5$ for $L \leq 105$.

Currently the best results for PSL values are known for $85 \leq L \leq 105$, and are reported in [6]. The optimal PSL sequences with $PSL = 1$ for $L = 2, 3, 4, 5, 7, 11$, and 13 are known as Barker sequences. The optimal binary PSL sequences up to $L = 74$ are also collected in [8].

Another important measure of smallness of AACF is the *merit factor* [9], given by:

$$\text{MF}(S) = \frac{C_0(S)}{2\sum_{k=1}^{L-1} |C_k(S)|^2}. \tag{4}$$

The merit factor is defined as the ratio of the energy of the mainlobe level to the energy of sidelobe levels.

Roughly speaking, there are two versions of LABS searches in the literature: One targets minimizing the PSL [1], [10]–[12] and the other maximizing the merit factor [13]–[16]. A sequence with the optimal PSL usually has a merit factor which is much lower than the optimal merit factor, and vice versa. Owing to the practical importance and widespread applications of sequences with good autocorrelation properties, in particular with low PSL values or high merit factor values, a lot of effort has been devoted to identifying these sequences via either analytical construction methods or computational approaches [17]. In this paper, our goal is to search for long binary sequences with low PSL values via a computational approach.

Nowadays, a parallel computation can be applied to tackle hard optimization problems. The power of several computers that are not necessarily placed in the same location, but can also be spread overseas, is joined together in solving real-world problems. The grid computing was used to perform computations for finding (binary) sequences in reasonable amount of time [13], [15], [18], [19].

In this paper, we used a stochastic algorithm for searching binary sequences with low PSL values. The main contributions in this paper can be summarized as follows:

- A new stochastic algorithm for searching binary sequences with low PSL values is proposed.
- A fitness function that can guide a search process toward global optima.
- The new best-known PSL values are obtained by proposed algorithm.

The rest of our paper is organized as follows. The background is given in Section II, where related work is also presented. Our proposed algorithm is presented in Section III. In Section IV experimental results are conducted and a brief discussion is given. Finally, the paper ends with a conclusion and future work in Section VI.

II. BACKGROUND

One of the main challenges when solving the LABS problem using an incomplete search is how to implement the calculation of AACF (Eq. 1) efficiently. Some researchers developed an efficient implementation of the AACF calculation [1], [2], [10], [14], [15], [21].

Algorithm 1 Algorithm for One Bit Flip of a Binary Sequence [20]

procedure Flip(f, S, Ω_S, L)
 $\delta_{min} \leftarrow \min(L - f - 1, f)$
 $\delta_{max} \leftarrow \max(L - f, f)$
 if $f \leq \frac{L-1}{2}$ **then**
  for $q \in [0, \delta_{max} - \delta_{min} - 1]$ **do**
   $\Omega_S[\delta_{min} + q] \; -= \; 2S[f]S[L - q - 1]$
  end for
 else
  for $q \in [0, \delta_{max} - \delta_{min}]$ **do**
   $\Omega_S[\delta_{min} + q] \; -= \; 2S[f]S[q]$
  end for
 end if
 if $f \leq \frac{n-1}{2}$ **then**
  for $q \in [0, L - \delta_{max}]$ **do**
   $\Omega_S[\delta_{max} + q - 1] \; -= \; 2S[f](S[2f - q] + S[q])$
  end for
 else
  for $q \in [0, L - \delta_{max} - 1]$ **do**
   $\Omega_S[\delta_{max} + q] \; -= \; 2S[f](S[\delta_{max} - \delta_{min} + q] + S[L - q - 1])$
  end for
 end if
 $S[f] = -S[f]$
end procedure

The time complexity of the trivial AACF calculation is $O(L^2)$, and the Fast Fourier Transformation (FFT) approach has the time complexity $O(L \log(L))$.

Recently, Dimitrov *et al.*, in [20], applied an efficient mechanism for single bit flipping calculation which is presented in Algorithm 1. The mechanism uses two one-dimensional arrays, S and Ω_S, to store a binary sequence and its sidelobes, respectively. Algorithm 1 performs an in-place memory update of Ω_S, when a single bit on position f is flipped.

The genetic algorithm is presented in [22]. It generates some offspring by the mutation operation (one-point or two-point mutation) and others by the one-point crossover operator. The fitness function is used as:

$$f_1(S) = \frac{\alpha}{\text{PSL}(S)} + \beta \cdot \text{MF}(S). \tag{5}$$

where α and β are empirical weight coefficients, which determine the importance of PSL and MF in the process of optimization.

A memetic algorithm was used for the LABS problem in [23]. Only a mutation operator was applied, and the k-opt local search was implemented by flipping each bit of the sequence. The fitness function is selected as:

$$f_2(S) = \frac{\text{MF}(S)}{\text{PSL}(S)}. \tag{6}$$

The results were presented for $L = 71$ to 100.

Algorithm 2 Algorithm for Binary Sequences PSL Optimization [20]

```
BestCost, Cost ← F(Ω_S), 0
isGImpr, isLImpr ← true, false
while true do
  if isGImpr then
    r ←R(n)
    for (i ← 0; i < L; i++) do
      Flip((r + i)%L, S, Ω_S, L)
      Cost ← F(Ω_S)
      if BestCost > Cost) then
        BestCost, isLIimpr ← Cost, true
        break
      else
        Flip((r + i)%L, S, Ω_S, L)
      end if
    end for
    if isLImpr then
      isGImpr, isLImpr ← true, false
      continue
    else
      isGImpr ← false
    end if
  else
    r ← 1+R(4)
    Q(1 + r, S, Ω_S)
    isGImpr, isLImpr ← true, false
  end if
end while
```

In [2], an evolutionary algorithm was applied to search for long binary sequences with low PSL values. Since the classic genetic algorithm is inefficient for the LABS problem, the algorithm adopted some features: Crossover operation was not applied, two-point mutation was used, the bit-climber was applied as a local search, and partial restart was implemented. The evaluation of the fitness function takes $O(L^2)$ operations for calculating $C_k(S)$. For each bit flip at s_i, $C_k(S)$ can be calculated from its previous value in $O(L)$.

Mow *et al.* [2] performed an experiment for finding which fitness function was most suitable for searching long LABS with low PSL. Four different fitness functions were used in their evolutionary algorithm: PSL, MF, f_2 (Eq. 6), and f_3, where the last one is defined as follows:

$$f_3(S) = \frac{1}{\sum_{k=1}^{L-1} |C_k(S)|^\gamma}, \quad \gamma \in \{1, 2, \ldots\}. \tag{7}$$

In [2], $\gamma = 4$ was used, and the experimental results showed interestingly that f_2 was an even more effective fitness function than PSL, even if PSL was the objective to be minimized. The experimental results for $L = 106$ up to 300 are reported, and for some chosen lengths between $L = 303$ and 4096.

In [10], an evolutionary algorithm is proposed to find binary codes (sequences) with peak sidelobe levels lower than the best known PSL values for selected lengths between 106 and 3000. Three enhancements were introduced that impacted the specific case of optimizing PSL significantly. One enhancement added multiple, weighted components of the score function, which combined PSL with two "softer" measures of sidelobe performance – average sidelobe amplitude and average sidelobe power. These components of the score were weighted so that PSL was the most important, but integrated sidelobe levels can continue with improvement for a given step in PSL. The second and third improvements were a fast-autocorrelation calculation and a local search which flipped every possible combination of up to 3 bits in the best sequence so far. The algorithm was run on a supercomputer asset, allowing multiple threads to run concurrently.

Lin *et al.* [1] recently published the 1bCAN and 1bPeCAN algorithms, where 1bCAN is used for aperiodic binary sequences design, while 1bPeCAN is used for periodic binary sequences design. The proposed algorithms are FFT based and, hence, can be used to design long sequences with lengths (up to $L \sim 10^6$ or even longer) on an ordinary laptop.

Yet another evolutionary algorithm, called SHC, is presented in [11]. The results are presented for sequences with lengths for $L = 106$ up to 300, and several best-known PSL values are reported. In very recently published paper [24] the author presents the PSL values for m-sequences for m equal to 18, 19, and 20.

III. OUR PROPOSED ALGORITHM

In this Section we present a new algorithm which is the improved version of the algorithm proposed by Dimitrov *et al.* [20].

An algorithm for solving long LABS problems to find low PSL values needs to be equipped with some important features:

- It requires an implementation of the efficient AACF calculation.
- It also needs a fitness function that can guide a search process toward global optima, which is not an easy task, since the search landscape of an LABS problem is very rugged, i.e., with many local optima.
- In the case of an evolutionary algorithm, a diversity mechanism is welcome, and/or a restart mechanism that can prevent the stagnation of an algorithm in local optima.

The fitness function in Eq. (2) considers the value of the maximum peak sidelobe. Many $C_k(S)$ may have the same maximum value in the LABS problem. On the other hand, fitness function f_3 (Eq. 7) considers all sidelobes $C_k(S)$, $k = 1, 2, \ldots, L-1$, but gives priority to the largest sidelobes. In the case when $\gamma = 2$, $f_3(S)$ is equivalent to the merit factor MF. In the case when $\gamma \geq 3, f_3(S)$ has a similar effect as $1/\text{PSL}(S)$.

There are two observations in the literature when searching for low PSL sequences:

- In general, a different tradeoff between the PSL and the merit factor can be achieved by choosing a different value of γ [2].
- Several authors selected $\gamma = 4$ [2], [11], [20] associated with the fitness function $f_3(S)$.

Our motivation in this paper is to combine both observations into a single feature in our algorithm, i.e., to use fitness function $f_3(S)$ with different values of γ during the optimization process. Questions may arise when to perform changes of γ and which values can usefully be applied for γ?

Our algorithm is based on the algorithm proposed in [20], which is shown in Algorithm 2. All the changes made on the algorithm proposed in [20] are presented in blue.

Both algorithms use the efficient one bit flip calculation that is presented in Algorithm 1.

In our algorithm we introduce a new fitness function with 6 choices as follows:

$$F_a(S) = \begin{cases} \sum_{k=1}^{L-1} |C_k(S)|^3, & \text{for } a = 0, \\ \sum_{k=1}^{L-1} (|C_k(S)|^3 + |C_k(S)|^2), & \text{for } a = 1, \\ \sum_{k=1}^{L-1} |C_k(S)|^4, & \text{for } a = 2, \\ \sum_{k=1}^{L-1} (|C_k(S)|^4 + |C_k(S)|^3), & \text{for } a = 3, \\ \sum_{k=1}^{L-1} |C_k(S)|^5, & \text{for } a = 4, \\ \sum_{k=1}^{L-1} (|C_k(S)|^5 + |C_k(S)|^4), & \text{for } a = 5. \end{cases} \tag{8}$$

A choice in the proposed fitness function $F_a(S)$ is selected according to parameter $a \in \{0, 1, \ldots, 5\}$.

A fitness function $F(\Omega_S)$ is incorporated in the original Algorithm 2 in Steps 1 and 8, while, in our algorithm, we use $F_a(S)$ (Eq. 8). At the beginning of the optimization process of our algorithms, in Step 1 of Algorithm 2, we initialize parameter a to 3, which means that we start our algorithm using the fitness function $F_a(S) = \sum_{k=1}^{L-1}(|C_k(S)|^4 + |C_k(S)|^3)$.

Actually, instead of calling the Flip function (Algorithm 1) in Step 1 more times in order to initialize Ω_S, for longer sequences we advise to use the trivial calculation of the AACF, and then initialize Ω_S which is faster than calling the Flip function more times.

The next change in Algorithm 2 is made in Step 8, which is needed since the fitness function is applied in this step too. Function $Q(x, S, \Omega_S)$ in Step 24 makes x flips at random bit positions in S. This function is applied to escape from the local minimum, when an algorithm is stuck in it.

The last change is also performed in Step 23, where we use $r \leftarrow 2 + \mathrm{R}(4)$ instead of $r \leftarrow 1 + \mathrm{R}(4)$, where $\mathrm{R}(n)$ is a function that generates a pseudo-random integer number $\in [0, n)$. Note that in the paper [20], the authors used $r \leftarrow \mathrm{R}(4)$, which is a small inconsistency with the source code, where $r \leftarrow 1 + \mathrm{R}(4)$ is used.

Changes between our algorithm and the algorithm proposed in [20] are in four Steps. In the next Section, we will present the obtained results in our experimental work, to see how these changes can influence the performance of our algorithm.

The complexity of Algorithm 2 depends mainly on the complexity of Algorithm 1 (bit flip operation with fitness function evaluation). In [20] it has been shown that the time complexity of Algorithm 2 is $O(L)$, where L is the length of a binary sequence. The main loop (it starts in Step 3) of Algorithm 2 also requires Z repetitions. The changes in our algorithm, that have been incorporated into Algorithm 2, do not increase the complexity, and, therefore, we can infer that our algorithm also has time complexity of $O(L)$, and in the case when $Z > L$, our algorithm has time complexity $(L \cdot Z)$.

IV. RESULTS

In this Section we present our experimental results. We used our improved version of the algorithm and the obtained results were compared with the best-known results of the state-of-the-art algorithms. The parameter a takes value from 0 up to 5 in this study it was set based on some additionally runs of our algorithm. We did not perform a fine tuning upper limit of this parameter. A description of the obtained experimental results is divided into the following parts, based on the sequence lengths:

- all binary sequences with $106 \leq L \leq 300$,
- selected binary sequences with length from 324 to 1936,
- selected binary sequences with length from 2000 to 4096,
- m-sequences with length up to 2^{17}, and
- sequence of $L = 10^6$.

A. BINARY SEQUENCES WITH LENGTH FROM 106 TO 300

There are some papers recently published that have reported results of the PSL values for $106 \leq L \leq 300$: (1) Mow *et al.* [2], (2) Dimitrov *et al.* [11] with the SHC algorithm, and (3) Coxson *et al.* [10]. In work [11], the authors have made a comparison of the best-known results against the results in [2] and also several other papers, and they reported the currently best-known PSL values for all lengths from 106 to 300. In [10], the authors gave some results for PSL with lengths in that interval. If we combine all reported results in all three mentioned works, we can see that the currently best-known PSL results for $106 \leq L \leq 300$ are shown in [11].

We run our algorithm for searching PSL sequences with lengths from 106 to 300, and our algorithm was able to find some new best PSL values. These new best-known PSL values are shown in Table 1, labeled as 'New', and

TABLE 1. **New best-known PSL values (New), compared to the current best-known PSL values (Old).**

L	Old	New	L	Old	New
115	7	6	203	10	9
116	7	6	204	10	9
117	7	6	205	10	9
118	7	6	206	10	9
119	7	6	207	10	9
120	7	6	208	10	9
121	7	6	209	10	9
125	7	6	210	10	9
134	8	7	212	10	9
135	8	7	213	10	9
136	8	7	229	11	10
137	8	7	230	11	10
138	8	7	231	11	10
139	8	7	232	11	10
140	8	7	233	11	10
141	8	7	234	11	10
142	8	7	235	11	10
143	8	7	236	11	10
144	8	7	237	11	10
145	8	7	238	11	10
146	8	7	239	11	10
147	8	7	240	11	10
148	8	7	241	11	10
149	8	7	242	11	10
150	8	7	243	11	10
155	8	7	244	11	10
169	9	8	245	11	10
170	9	8	246	11	10
171	9	8	247	11	10
172	9	8	248	11	10
173	9	8	273	12	11
174	9	8	274	12	11
175	9	8	275	12	11
176	9	8	276	12	11
177	9	8	277	12	11
178	9	8	278	12	11
180	9	8	279	12	11
182	9	8	280	12	11
184	9	8	281	12	11
196	10	9	282	12	11
197	10	9	283	12	11
198	10	9	284	12	11
199	10	9	285	12	11
200	10	9	286	12	11
201	10	9	296	12	11
202	10	9			

they are compared to the current best-known PSL values (labeled as 'Old').

In Appendix in Tables 5 and 6 we present the merit factor (MF), normalized PSL in dB, and the binary sequence. For each length L, a sequence is presented using a hexadecimal notation. We decode each hexadecimal digit in binary form ($0 \mapsto 0000$, $1 \mapsto 0001$, $2 \mapsto 0010$, ..., $F \mapsto 1111$), and, if necessary, remove the initial 0 symbols to obtain a binary string of the appropriate length. Then we convert each 0 to $+1$, and each 1 to -1 to obtain the binary sequence.

The results in Table 1 show that we have found 91 new sequences with the best-known PSL values within the interval from 106 to 300.

B. BINARY SEQUENCES WITH LENGTH FROM 324 TO 1936

We present the results of our algorithm for binary sequences with lengths $L = x^2$ for $x \in \{18, 19, \ldots, 44\}$, compared

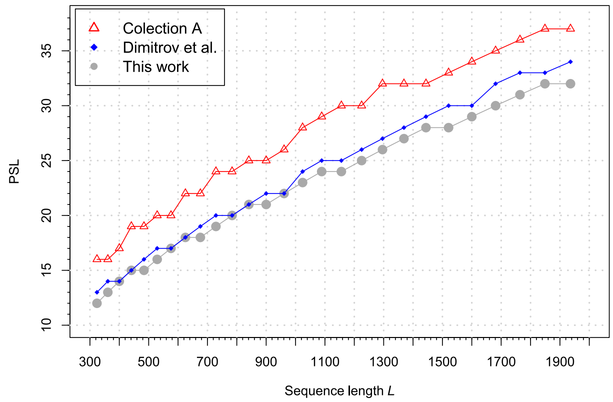

FIGURE 1. **Comparison with other state-of-the-art algorithms known in literature. "Collection A" and Dimitrov *et al.* are results taken from [20]. Lower values are better.**

TABLE 2. **New best-known PSL values found for some $L \in \{18^2, 19^2, \ldots, 44^2\}$, compared to the current best-known PSL values (Old).**

L	Old	New	L	Old	New
324	13	12	1225	26	25
361	14	13	1296	27	26
484	16	15	1368	28	27
529	17	16	1443	29	28
676	19	18	1520	30	28
729	20	19	1599	30	29
900	22	21	1680	32	30
1024	24	23	1763	33	31
1089	25	24	1848	33	32
1156	25	24	1935	34	32

to the algorithm proposed by Dimitrov *et al.* [20], and the collection (of the results) of the state-of-the-art algorithms, also presented in [20]. This collection is called "Collection A". The obtained results are shown in Figure 1 and in Table 2.

Figure 1 depicts PSL values obtained by the collection of the state-of-the-art algorithms, the algorithm in [20], and our algorithm. One can see that our algorithm found binary sequences with lengths $L = x^2$ for $x \in \{18, 19, \ldots, 44\}$ with the PSL values that are equal (in 7 cases) or lower (in 20 cases), and never worse in comparison to algorithm in [20]. On the other hand, both algorithms obtained better results than the state-of-the-art algorithms in "Collection A". The new best-known PSL values and their sequences, MF, and NPSL are presented in Appendix in Tables 7 and 8.

C. BINARY SEQUENCES WITH LENGTH FROM 2000 TO 4096

In literature [10] and [2] there are results of some larger sequences. We performed an experiment for searching a low PSL value on these lengths of binary sequences, and the obtained results are collected in Table 3, where PSL, NPSL, and MF are presented for our algorithm, compared with the current best-known PSL values.

For all lengths in Table 3 our algorithm obtained the best results, compared to the other algorithms, and it was able to

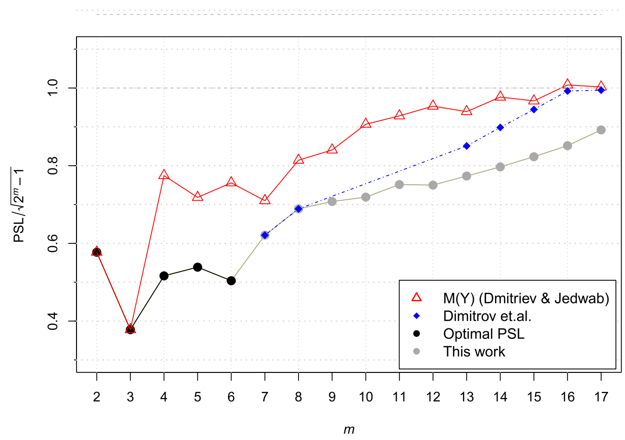

FIGURE 2. Comparison of the growth rate PSL with $\sqrt{L}$: We present PSL/$\sqrt{(2^m-1)}$ vs m, and $L = 2^m - 1$. The results for Dimitrov *et al.* are taken from [20], and M(Y) are from [25].

TABLE 3. New best-known PSL values, compared to the current best-known PSL values.

L	New Best PSL (NPSL) MF	Current Best PSL (NPSL)
2000	33 (-36.65 dB) 4.3026	38 (-34.42 dB) [10]
2048	34 (-35.60 dB) 4.2731	38 (-34.63 dB) [10]
2197	35 (-35.96 dB) 4.3928	45 (-33.77 dB) [2]
2250	35 (-36.16 dB) 4.4790	41 (-34.78 dB) [10]
2500	37 (-36.59 dB) 4.5598	44 (-35.09 dB) [10]
3000	41 (-37.29 dB) 4.4947	51 (-35.39 dB) [10]
4096	48 (-38.62 dB) 4.6129	61 (-36.54 dB) [2]

improve the current best-known PSL values too. One can see that new best-known PSL values have been improved from 4 (for $L = 2048$) up to 13 in the case of $L = 4096$, where the current best-known PSL value was improved from 61 to 48. The new obtained sequences are shown in Table 9.

D. M-SEQUENCES

In [20] it is outlined that the reason for the lack of publishing results for binary sequences of length greater than 2^{12} is due to the quadratic computing complexity of some state-of-the-art algorithms.

Nevertheless, we performed the next experiment to compare the results of our algorithm with m-sequences. Notice, m-sequences exist only for lengths $L = 2^m - 1, m \geq 1$, $n \in \mathbb{N}$. The obtained results are presented in Figure 2, compared with the results in the literature. Figure 2 shows a comparison of the growth rate of PSL with $\sqrt{L}$ for m-sequences. The optimal PSL values are known for $m \leq 6$. The values $M(Y)$ are taken from the work of Dmitriev and Jedwab [25] where the authors studied the growth rate of PSL values. We added values taken from Dimitrov *et al.* [20], and the results of our algorithm, so we have three lines on the right side of the figure, where the PSL values of longer sequences are depicted. If we look at the values for m-sequences between 13 and 17, we can see that several values are below 1 (only some values $m = 16$ and $m = 17$ are close to 1), and our algorithm had found all PSL values that are below 0.9. We are aware that we can not make any assumption about the growing rate of PSL for longer sequences ($m > 17$).

The PSL values for $13 \leq m \leq 17$ are collected in Table 4.

TABLE 4. PSL values compared to the known results for m-sequences.

m	$L = 2^m - 1$	M(Y) [25]	A_n [20]	this work (NPSL)
13	8191	85	77	70 (-41.36 dB)
14	16383	125	115	102 (-44.11 dB)
15	32767	175	171	149 (-46.85 dB)
16	65535	258	254	218 (-49.56 dB)
17	131071	363	360	323 (-52.16 dB)

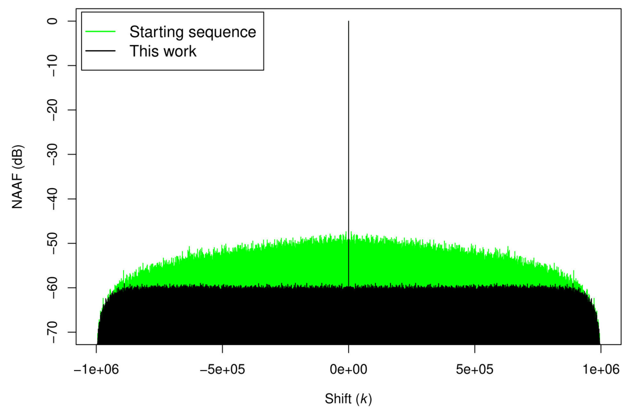

FIGURE 3. The NAAF (in dB) of the sequence with length $L = 10^6$ obtained by our algorithm, PSL = 1125, NPSL = −58.98 dB.

E. VERY LONG BINARY SEQUENCE

In the last part of the experimental works we made a comparison of our algorithm with the state-of-the-art algorithm called 1bCAN [1], which is FTT based, and, hence, can be used to design long binary sequences. The comparison is performed on a sequence with length $L = 10^6$. The 1bCAN obtained an NPSL value of −56.1 dB and it is about 9.5 dB lower than that of the initial sequence.

Figure 3 depicts the NAAF in dB, i.e., $20 \log_{10} \frac{|C_k(S)|}{L}$ of the sequence obtained by our algorithm, where the NPSL of this sequence is reduced to −58.98 dB (which is 2.88 dB better than 1bCAN) and it is more than 12 dB lower than that of the initial sequence.

To summarize the obtained results in each part in Sections IV-A to IV-D, one can see that we have found many binary sequences with new best-known PSL values, while for the sequence with $L = 10^6$ we got the better NPSL value compared to the 1bCAN algorithm.

V. LIMITATIONS OF THE STUDY

In this study we used heuristic algorithm for finding binary sequences with low PSL values. The obtained best-known PSL values are not necessarily the optimal, and for longer sequences the obtained PSL values are pretty surely not optimal.

Our main objective used in the proposed algorithm was to minimize the PSL value. Based on our best knowledge

TABLE 5. New best-known binary sequences and their PSL values (New), compared to the current best-known PSL values (Old).

L	Old	New	Hexadecimal form	MF	NPSL (dB)
115	7	6	718E250B1AF44181F8A8EEC96DA9B	5.3629	-25.6509
116	7	6	4E77C15A768EF42AE36F850482899	4.5034	-25.7261
117	7	6	1AEC5AD5F3C65E9ECD8319498FF457	4.3931	-25.8007
118	7	6	345A27A75995D48F193A621A400B3F	5.0050	-25.8746
119	7	6	30C8B8DB39902EFDD2FE2BE18B5A5C	5.0756	-25.9479
120	7	6	E6A898F37D681069410BCD94EC1F41	5.0139	-26.0206
121	7	6	13A2F87E39F577625B214D08DFEB149	5.2894	-26.0927
125	7	6	04B1CA736D584AAA342BFF0ED8FCFEC9	5.5883	-26.3752
134	8	7	362C276FC7968C12B62A9DD167BC57C37D	4.6398	-25.6401
135	8	7	19184C0E9774FD6DA533EEA61CE94FD0AF	4.7486	-25.7047
136	8	7	FC4107F53343304B242D0B2F4CF5538A86	4.8368	-25.7688
137	8	7	142D3D0BCF6D38A44F93FF174CEEDC65547	4.7301	-25.8325
138	8	7	1CE9CC6D4B14EC7F5DDBAB832C96FB040EB	4.5845	-25.8956
139	8	7	76BB52A425FA6E413BA8A3ECE79F7CFE4F0	3.8565	-25.9583
140	8	7	4BD6B9C7F977B394E4F7C3277418175FBAA	4.0867	-26.0206
141	8	7	04410863ABADCB894DAECA959BF0E17F06FC	5.3966	-26.0824
142	8	7	37234B0DD492E0F9D62554E7E410CF96EFFE	4.8216	-26.1438
143	8	7	20DC956CDC8C0FEB5B183EAE71CA09300165	4.1716	-26.2048
144	8	7	A296499A288EF9403E15E918903138AD6F2F	5.1429	-26.2653
145	8	7	125DB784263CAF323A54E5C67E81506021361	4.4469	-26.3254
146	8	7	1A331AB138FA2A0857DE19F7E2DB0DA46D92B	4.5295	-26.3851
147	8	7	3399E74668514B98BE80D892AFCA001E16092	4.4118	-26.4444
148	8	7	1D37553C0A652AFC590CCE804053C62D6D19F	4.9557	-26.5033
149	8	7	0DAE2AA951EF1A6583FB408581203732F61389	4.8644	-26.5618
150	8	7	285E9B09F1926EF9D6BFBCC804E3AE82D613D5	4.7690	-26.6199
155	8	7	12D4F30FE22A5AEC49F750C186DE45C7EBF774B	4.6759	-26.9047
169	9	8	1243329B5DC4BFBD48798D317478500F157B046822E	4.2552	-26.4959
170	9	8	1E6F95C9FC4AD691DFDBA00EC24D1D3638A39F0AA98	3.8217	-26.5472
171	9	8	4F780C417D388D543859D38C4A9626E7E97DF116CB2	4.8460	-26.5981
172	9	8	B0A0FA09DBA397644BAEFC09C67D8E5EE7B79DAC948	4.3893	-26.6488
173	9	8	181044A8285F9B9A1F0C6177697D04335AE9D21976D1	4.4143	-26.6991
174	9	8	055B562DE7AAD61933D9B8180C37183EE0DA11929615	4.1829	-26.7492
175	9	8	771810888E2F5BC653CDBA5597BA5791D33F241F5879	4.7628	-26.799
176	9	8	F07C62623A097331EF2C67FFC9124ACD35FA1552C5A2	4.4251	-26.8485
177	9	8	16257F74B5D058DAFCF99E73588E964580B83EE1468DD	4.5378	-26.8977
178	9	8	0F74A2147F85E4145256952FF22DBE311C0C1A2265C64	4.7023	-26.9466
180	9	8	9FCBFED27A1D61A66262E753AE189366AF0CC7B50147F	4.5737	-27.0437
182	9	8	20CF6297023CEB5471BB6C5659D65E9F81A2D64B7F7727	4.7279	-27.1396
184	9	8	3693A99269D546ED0B96B00EACB4F70A0C0FFEE79CCFCA	4.6454	-27.2346
196	10	9	DFFA5BE78D1557DE953E0ED323D22684C86DEE634F11EE722	4.6599	-26.7603
197	10	9	006C4B21337EEB69EE7583B287BB77833978EAF6854F845761	4.3294	-26.8045
198	10	9	01ADEE86D7EECBFED391945530EE17BC11C1A976CF19B11EE5	4.0914	-26.8485
199	10	9	557969CA505CF6779B51D8307959B9A71BA7FA0FA0449DD102	4.2793	-26.8922
200	10	9	B7BA746EC57FF88F60C2F39C6CC696872635937ED3C1574520	4.4763	-26.9357
201	10	9	1DEBDB52CC908256E1C6CA4865EA8D9759147D0FC0C07311BA0	4.6934	-26.9791
202	10	9	0B27982C9E75450BF3A438196A843A084FACCD89A803D0CED79	4.4928	-27.0222
203	10	9	5B3047F11C07415860CA0B9E49ECB1D53AE4860B5F9D9E8D2A4	4.5575	-27.0651
204	10	9	05DA4A6B92E9AFE5C5AFC7739528500F7DC789DC59B103ECD95	4.3081	-27.1078
205	10	9	129126BB35FD904E501AAB3B087CEFA98E1E0F41A830615A3160	4.5188	-27.1502
206	10	9	20279047EA010198957BD4E4CBCDA34178DD9254D7943A1AC396	4.8102	-27.1925
207	10	9	22FF08858C50880CACAA71B0BF34E4816C7932E8541B0E6A7D81	4.2926	-27.2346
208	10	9	754B660D31E0CDD06D2E55370C6E0839169243FA9FBFA8413DD6	4.4805	-27.2764
209	10	9	1454D1AD0858BAB378A30CE8312030BED8039E8EC7B4DE042BA0F	4.6272	-27.3181
210	10	9	350D53371277256D1E389B6E0585F5A607A8B75828CF47FF7BBC4	4.7186	-27.3595
212	10	9	EAA19E5581A32142F278A686689C202269EFEC2563B335C97F851	4.8038	-27.4419
213	10	9	1D16A857E099ED161CE1A1899195D5964E7FDF10862D83B5EF4B21	4.8368	-27.4827
229	11	10	073722F28CEAC769E47775BE4F2FBF52987E8893200D4FDA2EF9414A0F	4.1953	-27.1967
230	11	10	14252075A3A22505676A921A3E236CA0FD8C1E7BDD2E936221859862DC	4.1542	-27.2346
231	11	10	7B357A90B17A80DDD566FBF3D81A4FA97C187186598C571993427034F6	4.2196	-27.2722
232	11	10	B2EF23105818A7FF316ABEABC8E9C2D700DA64B470963640CAC51AEC5C	4.4943	-27.3098
233	11	10	070D802703C330459C3E60C445DB1EFB15D52954D2E9B08AD6EA52DFC42	4.1658	-27.3471
234	11	10	2962DE8D9ED0484D9C8A207E2C06080A1BE87B69CAA6CD31473819C656B	4.3896	-27.3843
235	11	10	4C6EFED25DB58E900CD728FD8FCDD2B70E9EE7E1D34620DF2AAA0D882F3	5.0415	-27.4214
236	11	10	72477C8E7213B49D59953C7152237CFD20AAC3D2785F2E4A3F480124422	4.5047	-27.4582
237	11	10	192588150DE6A0C255638E4FCBCC553D45485B4B972B30F6E103F80CE932	4.3167	-27.495
238	11	10	100650322CDCA39233DF71E87211DBC189ED5A05B229A2D37A38F2C2BC04	4.4122	-27.5315

and experiences with optimizing the LABS problem we can conclude that the sequence with good PSL value has not very high merit factor, and vice versa. Therefore, the reported merit factors for sequences in this study are not the highest

TABLE 6. New best-known binary sequences and their PSL values (New), compared to the current best-known PSL values (Old).

L	Old	New	Hexadecimal form	MF	NPSL (dB)
239	11	10	78E3DCFF91DABD71B48A43C9D1D7DABA1248DB345525EC565427F30DBF88	4.6192	-27.568
240	11	10	1317FDB07FD98131986E50640A343D68BC865C78938072C35AAA1EA69451	4.6124	-27.6042
241	11	10	1EC9181AFEC85566F28FA31935E54F15CE21778C9F8483396BDBFCBF378B3	4.1653	-27.6403
242	11	10	06C53375E39D2572BCB11D0FBB686944BEB1FBADD711B43E6228F24017F85	4.3776	-27.6763
243	11	10	77A528BF79833A7A9749E9E99CF037CAA4DA40E5038A8B89DFF97540C1337	4.3057	-27.7121
244	11	10	430C78C752A279C5C1B911F87D5F5A016031E882894ED013F6CDA24A44259	4.3305	-27.7478
245	11	10	157714505FCED99BE76E5169FCF6ED68470788AC7C54AEFD286D306B907B08	4.8737	-27.7833
246	11	10	3A9611D0BD460AA2A54878046F96124BB30B9E4CCF919FE71C08513C91FDB5	4.4399	-27.8187
247	11	10	293D7BAED7C945A4409715972021E7C998A7CE92617C0BEA3BA466729760E2	4.0058	-27.8539
248	11	10	1843ACEA2A3B93E4E40D6C018B970138D0BD72EC9617763EB6D73A11622DB5	4.2926	-27.889
273	12	11	124F184101FD613D8004513C2D0C1F793B62EAEBD45EA5E84689D0ECCD56F66A6192E	4.3412	-27.8954
274	12	11	20BF07E48B47408D6E1FDC80035DC8A8C731435C48C29D7879494744464D36A5F2E9B	4.5069	-27.9272
275	12	11	3A32A2EFB411731A0DFCD52C6031C1E4EF9CDE12CBC552805198F21409F24C690ED8D	4.0724	-27.9588
276	12	11	8081B7B44DAC7DD7D6228596597C4743A02D12CA970A083C803E552B11AF33833BC66	4.1176	-27.9903
277	12	11	0FFB6E6FEDB5F002DF15F635146D5E8EE332271470F5E32296E546A3D9A4D97B0CE8C3	4.0787	-28.0217
278	12	11	3755F1099BACFE40B060E829B81A7FFA99CDA9952BBDAD748E2B7CD8E78B6870973EC3	4.1148	-28.053
279	12	11	7803E40F1692BD75E0AD3DD24399620169660C6995FDEE711915DDD9E8FEACD0E970BE	4.3020	-28.0842
280	12	11	6634B6C0BDC9A4E26E99466AD980A8E84253CAA36A63062950751DFEDF881E1BEBA4C3	3.6981	-28.1153
281	12	11	1A969013E34B960EFDED879FDBABC814052ADDB9C86AA2E2B3C833337D0CC35922CF51E	4.0668	-28.1463
282	12	11	30304F9668D29DA239F188A90F58FC464551886048C47A993C145B5DB2FED95B4101A7E	4.3084	-28.1771
283	12	11	26F5EE18B1055CC5D4AD5E84A2B05D080C94BC9820F15607FC66E3794C790A6D9613049	4.6385	-28.2079
284	12	11	1CD6BC96C339C1C3EBE0C116D54C0A81E88B3024898F513A75AD55DB481D4001E64E776	4.2729	-28.2385
285	12	11	1003FC61E3FDF3BEC34A6F59FB04CE6B7CE86B92964CB47C16472AA3A3176AF2BE2EEAC4	4.3325	-28.269
286	12	11	2E2906C2D69E5691EEEA0495F088D8A17698CF6E3ACF8C182078F43C8154066474235893	4.2803	-28.2995
296	12	11	2CF7F87379978A1C6B05F279501F4908A908AA9D05EAABC9CB58D323300EC63DBB244896BF	4.3564	-28.598

TABLE 7. New best-known binary sequences and their PSL values found for some $L \in \{18^2, 19^2, \ldots, 44^2\}$, compared to the current best-known PSL values (Old).

L	Old	New	Hexadecimal form	MF	NPSL
324	13	12	E31F6A18336E70970153017449FD9294310FA409A7341556E0320C52F39B6CB57E3E0EC42CB6 71191	4.4915	-28.6273
361	14	13	1ED66BEDEB5D6AD81A3FDFAC259A73704CB80279728CFE701FD57655845855AE0B9669CC 4C4EE5E3EA175032486	4.0695	-28.8713
484	16	15	6E86FDF133F3098D081EB35D2934A85428693D8D7B8FB24BD3A0F8BB85B2EA9CCEB07DA CA80395D036B13E025DF779080A232E3031064C315449CE9C2	4.7702	-30.1751
529	17	16	1E24FEBB9A846C5130801910A07075366612C86DBE95C4628354C5D47FD87C8C71AA44549D BEF15E4CC0CA7EF106661B2F4B9C30B5ACB40DAB5D8769E5434F941AF40	4.3665	-30.3867
676	19	18	61C1B1CED155318A70770828C248AD0BF4B46C4174E855F5AC9FDA6F58338771A3F62127FF A7DA02C01361533F189C6FB2E65F69BE6F2C2173C82282BF399583A6E4D87EB3AEF43954E5 95AFA1624D7C3107977CA	4.7163	-31.4935
729	20	19	0AAAB3C1C7CDB23BB82566F35FAB260F882CE90246EC7648B5FE375F32142B16522022536 D2F1CC3B01634EDBB04CCD483E61D68039E0BD3496849C48AC37B9291BF4769487539FC63 58BE47E5DCF3871142AA7B862AE7881091C0D	4.8760	-31.6795
900	22	21	DB06DD572C928569B9CC4D263C64D5F6B53C3CCB4697553762549AFCD575A774D418F03F F3A392B6C8CCCDCD0F2188AE7305ABF158FD528F55F85464B1EEBAC62C6A5B7004644E82 275E4CFF1EBE49EFFC9C97C059A70FF0930E74F5520E286B159FE281841900790CF2198FB52 C03822	4.6589	-32.6405
1024	24	23	340E83BEB9B177B3F79DDC9BB7206B450EAAA1B931E91FE2F51F92DF54A0B9CA06709DA AB0033D5F2471D348255B8C766727F7685CC6438B1A85DA8ACA284ED1359209B060829A75 E5601558DEB10AD4CBF6B1A61981EF2F5F226C067170E6159A79B934C079705434AE1EDBC 4799C4124D6D2777E704073D270D9250107B583	4.4501	-32.9714
1089	25	24	1B45BD113782480BCC35D87F7FED55781E8B064DA02834327D6C323BED046A9A417093DC 279CB4588B80DF9783F81FB7E7C746DB94335D943307CCC342510238F2D122B1BA04866D26 669E267D458795C8013310ED4EC0155AF0CE40B6F447D7996A9A470C7D5678E688EF68C544 2695F9C75A94E4AE262BBF5EF8A2CC126F4B5BE572A427B4AAF29	4.2654	-33.1363
1156	25	24	FD25BEC197F61F680B8A577DAB7DE9F2E91B26C9EAA1816C7D2A151286B6F2356776CAA6 8047656250D20EF9ACBDF408511841FB938656284591D9ED6710D08F0EB1EA6CAFBB5C18E0 63EE6B639D28EE7ED1E952CB079063524E04D342B11DB0E08F1C601CD7730B7267C0E960D0 905F462122F33709E3F548F97575DD0253DC750A47BE2222340FA1120D9519360D7E0	4.6436	-33.6549
1225	26	25	15523967287E77F06B7A95EBED988000C84F3B5A703C132F6BC8150FA0CCE997006D1CD3BC BADAFFAD0512F2EF9ADE43F01938E6EA2BBFFA124942E51755B15D817D61EC66DAB0B590 4376D8DD58EF1D266B0ACD14D78C64DBFCDBD7B14E0D9A45C27F3C5A47139EC33B23BB FD543DF3C7C5D1CAB979660C7E53C0404F9A189BE177BD15972CA957B299BB21E91508FA 203970A381E3C5A855E	4.4757	-33.8039
1296	27	26	11181A71960FE18597FA39D6C8A2FC355DCF8B79985883B258F6C76629BDA823DCF8E40EC 5E627757FA182FB54D424DAD418EA99E3AC01E542964142AEED6C47F8A56B65ADC3892803 C21699AA5A19071EA026C1B5BC7A22F617874274CC39B04DF755C97765FEED7DD26FC0C2 19EDD92F4B5DAE5C0C647F58F941662136729759EC691C76FD470D0D9A2E57AF23BF9E8DF 47385DB8162A4DEF938A4028B7DFF1595	4.4701	-33.9526

TABLE 8. New best-known binary sequences and their PSL values (New) found for some $L \in \{18^2, 19^2, \ldots, 44^2\}$, compared to the current best-known PSL values (Old).

L	Old	New	Hexadecimal form	MF	NPSL
1369	28	27	1CDEB6ABD22C3A280856FD73771B9F10C315333C5FEE1DDEA85BB13130DC82BFAE40FBA 8CE3B5D82B83E18F2D6557DEEFC09FE1B2F72AAB55C5EE5F31F4CE6FD851A449AAC25A5 1ADB2B809495C9A8B9DE0B297199DADA9EE8668F03308ABB39C019F74BF04BE2EAB36E05 26A3D482C91FA48185A5A93B4CF4391E3D3E7A3FEF6FD848B56F656A1C2C7127A645D593B 75279949869857E698911276FB67C3C4BC203F7CF006411E2309AC1A	4.5383	-34.1008
1444	29	28	67766AC04F862ACA0051D9C6A0F0CD426E80EFC145EF5ECA038807C31E81CBCB5397F810 16DF06325D6C2259C24F9192CB9B7E22A0BFC422F7C0B5074935249398DB4BFBFCBEADE4 46592BD8AC63F8C91397671788FD72518BCDBEA689986656E640E8D959BC669FE24894850F 2BE37B3B155C54D5C6EB6668DA08E44DBAE3EC7A3CB35EBD14AAA2CD2522163B17FBEC 778D5B29475FADEDD0C7B4EFD4D2841A47D448940F44E0F0F56CF75073AB5B39C709665D7	4.6282	-34.2482
1521	30	28	1CB5A58E1140FD01C6D32744132A64A3B7B9946C5BA569BD69BCD3DB62F4175B8D7C65B B5C454809A478B739F2C547CCAFA7BF33E1171A175F2D784FFD791E580D5763B6462EE65A FF688697F72B65870900DA9A22B1FD8E1770EBE35E3F78A1B323FCC3C9BA852E6C33DBB2 02A68FD5C76A6F8E446F8409414AF50A0707C142275C5BDA0149F65C06E8267F87401ADD2 CAACCECEBB93DDF7F86E50511A792F4B6FC6F24CE23B58C658F367715D206658A533C837 06BFBD470A82A32141F2	4.5068	-34.6994
1600	30	29	009FF36C1AA6AC72725977D59745D50A5FEFD51451BFAAE05C0B43C885552A1BA70E16D5 B53AAB7FE09ACA104AD7C63721352483C495F4CA469589FABD7ADCAF0547B7AD4861C06 E7436D2D371E474617C4C1BD18E93D80A15C48A8775F6954C85A6B634E915D766696172619 0BB8CB29AFBFDF71D2630E491381879E6583D8BD1495B8B8C8C58A111830238CFA21C9889 3A9DB4D6305AD3E0B67A08D37A2673410AD9C3789B7084713A8A6137E0E7D8F9C7C0F759 3CEE609BC3FDCFAF31DE34CDF9A04FA19F4DF6	4.6603	-34.8344
1681	32	30	172343515707D2169214184D37DAB4F02042AE27178F43956012C12229EF4BE09292C6870D28 E806155E7DD582BCBCBF97E5F5BDB854EFA44AC7B1B8331E1584D31DEF4EA5FE8DAAF7 87D2499354E36ADBEBC2BAA876428847B11B4CB39C8DC8834CBBFCBE46CFAE9D92D7EE C2880C6F6D23B656AFB26CE45CE0107E1EB338DB42CCC89B6721FA21D5117AE835F7DD08 FFD4E9B7724480FEE2840CD9CF3EA8995A5FB4F09A11CB1F64087C6548CF91AC66831F289 CEC66B3F383BE9F3FC94F20E38A46C1B46E9C6DD3988F77031973C0EF340	4.5354	-34.9689
1764	33	31	0D484ACB31BCDEC1D7F2DD31E01F049FF17309F8E6ADCFD8380034D996F21B7F86AF0188 DC8C3DD1FCB69CFF3E4A6719AEFEDCC8AD13C3B6F48E3DD4F641798CE4A7AC80985896 7299D1628E13760D2F27459EDE3E7736308D519FE9DB969342FB6E85417B25E1D8F4E816A7 604B78B3B1D14185EAC223A923DCF182CD74912C27EA47CF8266B46E20ABA5D25D7AE8E BE37E560D13DDE7120525E64B0FB6A2D55FEB35D65E9851A15ADD52465DC0CF9EC6FD74 FB7E4505D481AAADC1947182A528A73CB933811A63F17A89F75D5255165BC74515684002D C7873C50EC	4.3913	-35.1027
1849	33	32	0FBE7F16FD4DE6DF9B3E74A2D9D28ABB103D54D7961208A1F6A2BC6B60791A5F664F797F 2C8D098A139B66F388A886B8A7B7EB88A57C56A4251AF18380E020654B9C8296B03CBBF5E 17091E342CC746D01198639E8F4FE8EB66ABED92F523B84DF192543E87AB7617A9C2FD8C15 698596C69D2DD346E006F56EB74816C735714D19368390EEC55D848F5192BE76BE065364BF9 9C898F5CEE0EA27B3C6439B62A7CD70AE3E447F9D7E64DCB66B98D3D2C51A5FCB24C8EE D5918CA99A5788A88C94656C8FF6DE96DEDC4825E01167FDBC5D7FE968D04207FB79853EB CE3F711A2E74B7972E7D2E24F	4.6599	-35.2357
1936	34	32	A05503A93F027C13E43FA3C7C76B8E9ECFEE2EA31B835CAE18727A6054D76CA8EB62DA83 CFDB5C26C971D47DAE4681A3CDAAE8E09027F9D6E8D6B91C998F178FB553A1E2685ABB6 4A8CABC29B1B744656F641B30A5E80D7D64B14FB17C2E96B9F8853FFB464BBCF62034C171 A9DE7B75D0632EFEC09EE631769E35B5FAA452E70126631D257B7F39B65D43C399DB08C8F 39FEEC5BFB287FDC0E7EA9AFDED4B3B26E6FCE234B13DB00644694C0820DED5D67821247 60104EB1675B23D033EEAD7A94E7C0143726A6144A095B54CF25B3D2BD661B8D05E4762AA E525B32A8DA26BFC7DF8B2A383001AC379ED8E306322DD2365	4.6211	-35.6351

(see [13]). Nevertheless, in literature, some works applied both objectives (minimize PSL and maximize merit factor) simultaneously in the optimization process, like using the weighted sum of both objective.

VI. CONCLUSION

In this paper, we propose a new stochastic algorithm for searching long binary sequences with low PSL values. Our algorithm uses a set of different fitness functions and the active one is selected during the restart part. The experimental work included searching for binary sequences with low PSL values of selected lengths from 106 to 4096, and up to 131071 for m-sequences. The obtained results were compared with state-of-the-art algorithms and our algorithm found many long binary sequences with new best-known PSL values. In the comparison, performed on the long sequence with length $L = 10^6$, our algorithm obtained a slightly better NPSL result than the 1bCAN algorithm. As a future work one can see (1) using graphical processing unit implementation, and (2) searching of long binary sequences when MF is considered as the main objective.

APPENDIX. SEQUENCES WITH THE BEST KNOWN PSL VALUES

See Tables 5–9.

TABLE 9. Sequences for the PSL in Table 3.

L	Hexadecimal form
2000	3E377E599EE4B5B8AB04D26BF7CD7CBA88B075AF6CC18A925DFD97E077868C93652C68C8BB6EB1A664F98AAB01B81D28CAF39 AAE8C6EBC2E304F60A6F1A372C9FBC3E398EC71F754654AF20530D7C30C86A15F2B443E6A544015A99AC1867593E6A0F687E048E BDAB52C938A21BFD5273570E296713CF1C3CE24DE4A69E598623D8062FE9036EC2014C741A339A48DF11B36DD0D2E827D2F7FB7 E7DB02242BBD52A47636D011C30338BD3F8C2EB0B5A211BDED13996BFE51BCEB77F9B4171812D6E2600B2E3BCA7043C4FA6F42 8D67EEF064012001EE861636D2A1D51781E272F5712ED84B0F0980EA42F8281D88C5564D5F2063BB49DA7F2F76
2048	D2048F5A5661E63D0CDB612305A2BA5FDB3BAE9A0510EAB694FA2A612BA0CCE3F6E8B2CEEDE3CF46825141DF175B212758A7A 33D493224F2231EE96A52E40E99CEF7E78967CB0D61283F23DB3A59721E8A7187D8E53A896329FA0F9A9497E6292D5263301CEE0D A15AF5B4A39BCDBFBF0C7A2D5E152FA710A3C12648BC91117F7353B41A8B9EC669EC3275EF3185B549FDF8C05DF97D0D46F42E 6292D81E10C4F850DBD78CEDEA3F36EC9E361EE567966EA163A15F5E8F88221D6B7C87A2477EE4897D8169AF357E2C4C9883F7B9 BC0BED578CDC777C019BBFAF9E8F7DBF33E360A0B4845D5479BBABB8167A577500600C0852E7498B982349460943C9EDD644
2197	0E844B52C3CD611460EDDA1F52CFACA33B51D4198352C2DB6AD5B3AB9EFECFDA189FF78875BA7616537B7636CC7253157A3D46 9AC59792F611ECF32E26E6D8526DB937FD4B5B8D3BFDFD36CF9B501103E5F1530683972B3A010F2E1121AAB549DA3C08F2332E1B 971FECB3813BE10178659388E6F3E6A82EF2489986625FBDD0C1F8BD8EA3327573C698BDE6F2AF79251648124AA6A45E0F0FF409F 879DAED6EC1E04157191AFABFAEF88218422FB695406D6AF9A47F23B4B7279DA7065EBC380D3F927553109AC3BB538418E8CBFB E94B5ACB1177EAE516ED87BD84BA8A60A2EE4E4A6E2395A625CA1BF5DEBB41DE05672D11E8C03497FA2B40CE46176F95F719A6 D7258C0E3D5F83711A6A8B879F0CFEC2DD
2250	0C23529719DE19B030346E1EE991CCA9DD8E21E25FCA52C6AC4BE52B72D95C9EAB5DA2BCD6583C58AF31B15E8B2DD5718D3D DFFA21134395648A51CC937A7F2F54A61536C2EA9A37FE3570442F27C433B0BF3F2014D31B2A591F4EFA8717410AAD09B95776A8 A3042D16DB52541819D009BAACB658FD98FA07D387A7FD8E0110AD67295F61E19953B62CC87711A52192A7BF84E84B7F537FFA5A 0D79DC05916081C9D20B659B160EE9F7175EF83BCF85D3FE76CD3B58160DBD6806BE121E4423FA58A62EB0B84CDF30A3003CDD 1A603AC8E59F1CB6771F819797946246FE26B4C3444BD03FC906885BA29F8AA6BBE9249E1D70706132DCD4E75D3DB8CE2CE008F 202BF82A2FBB0A5CA1C009878109EC28B0222BEE3034330524
2500	FE4C337B1673FF68EE64C85C7E75B68C5C4F364DDABFE7BC5DCF365F76E22CA3C70DEED8E7BFC975DE8B681AA0E40CDEB0E1 C6466F397BA46D4CBA43BF2A47833E0A715033B3C76610843F22A6E41C15C19B8D55EE3773289FF882F2C1178F90C106F66F84CA8 2DD323F496B69416D63A4945D3E931015C351151A296F1558CFCD7C8D32C8F1F0A49E04AC8D43AC77A3AA47DA0A45FB2654EED EA847CD2BC1098C72100FFEECFEFCD82115A3B4B4B81A3842A2F28122D6E769D205A98376B95BDD8DA6328C4CC6BD7A6D5A9F 059BE8C0F9CEBA453B43828C87690A7B09173FD210E4EBAD0B406836AF7CE3364FA5EE216DC27B81BA39E14342F182D9ED4DF62 B0F18E370C972DBF0AFB321888C1B4A490514F2CB6A457561335C5E0FAADA2F1F83EE6A9D8BEAEA05D157744CD1AD3CA01D6F 366E83BA250CA
3000	F750E5A534E29B5F4F59517E39346A82BE9D19D7DF8E5B72CEB6C58E48138D972D3DE543F638D2D64A931CF283EF9D2AFF39A39 7EC5C41D88A89DB3A74B17975CA0E464334BFA560B17BB4F4ABDE39A23164D086D997B5DAB25D9F4616A465B978D5A64D24AD BB5737159BA6F78997673A443EAAFBC3330876796B3A29243D8A5901B20C79AF5A29B65DF8910F3A853D887487B8DC91B411B950 3F17B54C97F9A8EAC36778190DC01FDD611D8621421203FCE76CD85C17C847AF17348682CA23FFBFAE0010814E79ABACD63DEF5 30332FF5763E01F08C6D5425B598415EECDDC1A3D68FD9AA42CA4D2F97BF48F5FA5C309BE26874759C6B1122CDE077F4CA5B63 E4BBD14204C57295170FC5889C4D47D0EA8F80FDA626EA06F3843ABF5C7558AC46DC7E946BA68BE19DAAB80D9317A4E9BA797 619F813F20F36FE90803BBA04136B020580C59AE6660A2F8EFA79DE0AF6B063281D4202325A1F507690C4C1E82C8F0AACE6125709 7DB1183BFFEBD3D37E89DC567EB85F4
4096	3F1A35559CE9C8AB583A1281BE032FE062B43DB6860E047B72391444E64F1184424D449E2D318F60CA5A4B5E59480A4E72D52A883 AF66D2CA315A6EF623FA58769D47BE594C882C66F8DC195422D2A95E5B42F74E2F1B70ECC1D3308C04B59327B49623680948A0F4 6C0BCABDB140F843CE8C712011FB99FA5F5907BCE81531FD06610F0C9EB39E0FE20851A5C21F7003B7AA22882BB32D45060CBB7 6C6F1FE752E90848EC8E1CC72806357AA338C682269D8E17B745777BE0639ACADE450D4121398C650664443E1F0A8BC1F8B73979B B973C6457070F33D52B02BDF47FBE6A6D98B8F22D32E063147308F89E3A7B7EA56768FDDE5AFCF4E71322BF8AAB1806D11A2BB AFB58244558DCF950853E68327CFF9E835BEE4F995F245C9A95407FAB4540DF3D7ACC9965D900C8EF3014CB70A14AAA778253519 019458B9B43DAAA2FDCA34DE28C6E8A7FFB0D5B1ADFA30E68E0D2287486C06C14AFC8147ADF92CEC991B601FB283C1FD37A9 A6C9B45C90921E88BBDB6DBFACF462C3FF21A87AB9A2DB044C38CA34FFAF06E28F6AA08A4A224896D0E739DB1BB0512076D81 B5D39218A57588A0BDFD6795BF1AB44B269DE918D9AD196EE1AEC1E6335C6A4B343D15D64A2D535B9EAF336CC363C093159F2 AFC8644940F793BF477CA55110B8267BD36DA77837148469E3770BF2497801535997ADBE9F8C591EB2CD3D687E2B70D882E

ACKNOWLEDGMENT

The authors would like to thank Miroslav Dimitrov for making available the source code of algorithm [20]. They would also like to acknowledge the Slovenian Initiative for National Grid (SLING), and HPC-RIVR for using the computation power for performing the experiments.

REFERENCES

[1] R. Lin, M. Soltanalian, B. Tang, and J. Li, "Efficient design of binary sequences with low autocorrelation sidelobes," *IEEE Trans. Signal Process.*, vol. 67, no. 24, pp. 6397–6410, Dec. 2019.

[2] W. H. Mow, K.-L. Du, and W. H. Wu, "New evolutionary search for long low autocorrelation binary sequences," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 51, no. 1, pp. 290–303, Jan. 2015.

[3] G. E. Coxson, C. R. Hill, and J. C. Russo, "Adiabatic quantum computing for finding low-peak-sidelobe codes," in *Proc. IEEE High Perform. Extreme Comput. Conf. (HPEC)*, Sep. 2014, pp. 1–6.

[4] A. Dzvonkovskaya and H. Rohling, "Long binary phase codes with good autocorrelation properties," in *Proc. Int. Radar Symp.*, May 2008, pp. 1–4.

[5] K.-U. Schmidt, "Sequences with small correlation," *Des., Codes Cryptogr.*, vol. 78, no. 1, pp. 237–267, Jan. 2016.

[6] C. J. Nunn and G. E. Coxson, "Best-known autocorrelation peak sidelobe levels for binary codes of length 71 to 105," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 44, no. 1, pp. 392–395, Jan. 2008.

[7] K.-U. Schmidt, "Binary sequences with small peak sidelobe level," *IEEE Trans. Inf. Theory*, vol. 58, no. 4, pp. 2512–2515, Apr. 2012.

[8] A. N. Leukhin and E. N. Potekhin, "Optimal peak sidelobe level sequences up to length 74," in *Proc. Eur. Radar Conf.*, Oct. 2013, pp. 495–498.

[9] M. Golay, "Sieves for low autocorrelation binary sequences," *IEEE Trans. Inf. Theory*, vol. IT-23, no. 1, pp. 43–51, Jan. 1977.

[10] G. E. Coxson, J. C. Russo, and A. Luther, "Long low-PSL binary codes by multi-thread evolutionary search," in *Proc. IEEE Int. Radar Conf. (RADAR)*, Apr. 2020, pp. 256–261.

[11] M. Dimitrov, T. Baitcheva, and N. Nikolov, "Efficient generation of low autocorrelation binary sequences," *IEEE Signal Process. Lett.*, vol. 27, pp. 341–345, 2020.

[12] A. N. Leukhin and E. N. Potekhin, "A bernasconi model for constructing ground-state spin systems and optimal binary sequences," *J. Phys., Conf. Ser.*, vol. 613, May 2015, Art. no. 012006. [Online]. Available: http://stacks.iop.org/1742-6596/613/i=1/a=012006
[13] J. Brest and B. Bošković, "A heuristic algorithm for a low autocorrelation binary sequence problem with odd length and high merit factor," *IEEE Access*, vol. 6, pp. 4127–4134, 2018.
[14] J. E. Gallardo, C. Cotta, and A. J. Fernández, "Finding low autocorrelation binary sequences with memetic algorithms," *Appl. Soft Comput.*, vol. 9, no. 4, pp. 1252–1262, Sep. 2009.
[15] B. Bošković, F. Brglez, and J. Brest, "Low-autocorrelation binary sequences: On improved merit factors and runtime predictions to achieve them," *Appl. Soft Comput.*, vol. 56, pp. 262–285, Jul. 2017.
[16] C. Günther and K.-U. Schmidt, "Merit factors of polynomials derived from difference sets," *J. Combinat. Theory A*, vol. 145, pp. 340–363, Jan. 2017.
[17] J. Song, P. Babu, and D. P. Palomar, "Optimization methods for designing sequences with low autocorrelation sidelobes," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 3998–4009, Aug. 2015.
[18] T. Packebusch and S. Mertens, "Low autocorrelation binary sequences," *J. Phys. A, Math. Theor.*, vol. 49, no. 16, 2016, Art. no. 165001.
[19] D. J. Katz, S. Lee, and S. A. Trunov, "Crosscorrelation of Rudin–Shapiro-like polynomials," *Appl. Comput. Harmon. Anal.*, vol. 48, no. 2, pp. 513–538, Mar. 2020.
[20] M. Dimitrov, T. Baitcheva, and N. Nikolov, "On the generation of long binary sequences with record-breaking PSL values," *IEEE Signal Process. Lett.*, vol. 27, pp. 1904–1908, 2020.
[21] S. Halim, R. H. Yap, and F. Halim, "Engineering stochastic local search for the low autocorrelation binary sequence problem," in *Proc. 14th Int. Conf. Princ. Pract. Constraint Program. (CP)*. Berlin, Germany: Springer-Verlag, 2008, pp. 640–645.
[22] X. Deng and P. Fan, "New binary sequences with good aperiodic autocorrelations obtained by evolutionary algorithm," *IEEE Commun. Lett.*, vol. 3, no. 10, pp. 288–290, Oct. 1999.
[23] S. Wang and X. Ji, "An efficient heuristics search for binary sequences with good aperiodic autocorrelations," in *Proc. Int. Conf. Wireless Commun., Netw. Mobile Comput.*, Sep. 2007, pp. 763–766.
[24] M. Dimitrov, "On the aperiodic autocorrelations of rotated binary sequences," *IEEE Commun. Lett.*, early access, Dec. 29, 2020, doi: 10.1109/LCOMM.2020.3047899.
[25] D. Dmitriev and J. Jedwab, "Bounds on the growth rate of the peak sidelobe level of binary sequences," *Adv. Math. Commun.*, vol. 1, no. 4, p. 461, 2007.

JANEZ BREST (Senior Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Maribor, Maribor, Slovenia, in 1995, 1998, and 2000, respectively.

Since 1993, he has been with the Computer Architecture and Languages Laboratory, University of Maribor, where he is currently a Full Professor with the Faculty of Electrical Engineering and Computer Science. His research interests include evolutionary computing, artificial intelligence, and optimization. His fields of expertise embrace programming languages, and parallel and distributed computing research.

Dr. Brest is also an Associate Editor of the journal *Swarm and Evolutionary Computation*.

BORKO BOŠKOVIĆ (Member, IEEE) received the B.Sc. and Ph.D. degrees in computer science from the University of Maribor, Maribor, Slovenia, in 2004 and 2010, respectively.

Since 2000, he has been working with the Laboratory for Computer Architecture and Programming Languages, University of Maribor, where he is currently an Assistant Professor with the Faculty of Electrical Engineering and Computer Science. His research interests include evolutionary computing, optimization, natural language processing, and programming languages.