IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 14, 2017, accepted December 22, 2017, date of publication January 5, 2018, date of current version February 28, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2789916

A Heuristic Algorithm for a Low Autocorrelation Binary Sequence Problem With Odd Length and High Merit Factor

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 lovenian Research Agency under Grant P2-0041 "Computer Systems, Methodologies, and Intelligent Services."

ABSTRACT A low autocorrelation binary sequence (LABS) problem is a hard combinatorial problem and its solutions are important in many practical applications. Till now, the largest best-known skew-symmetric sequence with merit factor greater than 9 had a length of 189. In this paper, a new heuristic algorithm is presented for the LABS problem. The proposed algorithm stores promising solutions and this mechanism enables the algorithm to perform local searches on these solutions in a systematic way. Our algorithm was tested on skew-symmetric sequences and the obtained results are compared with results of the state-of-the-art algorithms. The proposed algorithm was able to find some new best-known skew-symmetric solutions with merit factor greater than 9 in sequence lengths over 200. The obtained results improve the suggestion from 1985 (Beenker *et al.*) and 1987 (Bernasconi) greatly, where the merit factor is approximately equal to 6 for long skew-symmetric sequences with length up to 199. Now, the largest best-known skew-symmetric sequence with merit factor greater than 9 has the length 225. Additionally, now all merit factors are greater than 8.5 on the interval from 159 up to 225 for odd lengths.

INDEX TERMS Algorithm design and analysis, combinatorial optimization, heuristic algorithms, aperiodic autocorrelation, merit factor, skew-symmetric sequences.

I. INTRODUCTION

The problem of determining the maximal merit factor for binary sequences is an old and apparently very difficult problem in combinatorial optimization [1], [2]. Binary sequences with low autocorrelations are important in communication engineering [3], [4] and in statistical mechanics, such as groundstates of the Bernasconi model [5]–[7]. In mathematics, this problem has attracted sustained interest (see *Littlewood polynomial*) [8]–[10]. A recently published survey [11] reviews the state of knowledge of sequences with small correlation. Long binary sequences are essential for various applications of the coded exposure process [12], [13].

Consider a binary sequence of length L, $S = s_1\ s_2 \ldots s_L$ where each $s_i \in \{+1, -1\}$. The off-peak autocorrelations of S are defined

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

and the energy of S is

$$E(S) = \sum_{k=1}^{L-1} C_k^2(S). \tag{2}$$

The Low Autocorrelation Binary Sequence (LABS) problem involves assigning values to the s_i that minimize $E(S)$ or maximize the *merit factor* $F(S)$ [14], which is defined as

$$F(S) = \frac{L^2}{2E(S)}. \tag{3}$$

The merit factor is a measure of the quality of the sequence in terms of engineering applications [15].

The autocorrelation function is considered to be one of the most common methods for extracting various characteristics from signals, e.g., speech signals [16], where the autocorrelation function is applied one frame at a time in order to extract the peak and its corresponding lag.

Owing to the practical importance and widespread applications of sequences with good autocorrelation properties, in particular with low PSL values (see (5)) or large merit factor values, a lot of effort has been devoted to identifying these sequences via either analytical construction methods or computational approaches in the literature [4], [17].

The search space of the LABS problem is of size 2^L. To locate good (optimal) solutions, there exist two approaches: *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, is able to solve larger instances, and the obtained result may be optimal or close to optimal, i.e., it does not guarantee optimality.

The skew-symmetric sequences have odd length with $L = 2k - 1$ for some k, and satisfy

$$s_{k+i} = (-1)^i s_{k-i}, \quad i = 1, 2, \ldots, k-1. \tag{4}$$

The restriction of the problem to skew-symmetric sequences reduces the effective length of the sequence from L to approx. $L/2$, but the effective *size* of the problem from 2^L to approx. $2^{(L/2)}$ [5], but the skew-symmetric solutions might not be optimal for each problem instance.

One of the main challenges when solving the LABS problem using the incomplete search is how to implement a calculation of energy efficiently and researchers developed an efficient implementation of the energy calculation [18]–[20]. Note that similar efficient calculation can be applied to finding a skew-symmetric solution of the odd length problem instances.

In this paper, we proposed a new stochastic algorithm (xLastovka) for solving skew-symmetric binary sequences. The obtained skew-symmetric solutions might not be optimal for each LABS problem instance. The proposed algorithm was able to find some new best-known skew-symmetric solutions with high merit factors. Based on our best knowledge, it is the first time that a merit factor greater than 9 has been found for $L > 200$.

The rest of our paper is organized as follows. Related work is presented in Section II. Section III describes the xLastovka algorithm in detail. Experiments are conducted in Section IV to verify the advantages and disadvantages of xLastovka. Finally, the paper ends with a Conclusion and future work in Section V.

II. RELATED WORK

Roughly speaking, there are two versions of LABS searches in the literature [21]: The one targets low Peak Sidelobe Level (PSL) [22] and the other targets high merit factor (or equivalently, low sidelobe energy). The *PSL* [23] of a binary sequence of length L is defined as

$$PSL(S) = \max_{k=1}^{L-1} |C_k(S)|. \tag{5}$$

It is not possible to consider PSL and merit factor at the same time, i.e., a sequence with the optimal PSL has a merit factor which is much lower than the optimal merit factor. In this paper, our key focus is to search for long sequences with high merit factors.

Heuristic algorithms are proposed for solving real-world problems. A heuristic algorithm can solve small instances easily and performs reasonably when tackling larger instances to find fast and close to optimal solutions.

To tackle the LABS problem researchers have applied (1) Exact techniques, such as enumeration [24], branch and bound [5], [25], (2) Stochastic techniques, such as tabu search [19], memetic algorithm combined with tabu search [18], evolutionary algorithm with a suitable mutation operator [26], evolution strategy [24], genetic algorithm [27], and directed stochastic algorithm [15]. Recently, a self-avoiding walk technique has been applied in the lssOrel algorithm [20]. Memetic agent-based paradigm [28], which combines evolutionary computation and local search techniques using parallel GPU implementation, is one of the promising meta-heuristics for solving a LABS problem.

Currently, the optimal solutions for even and odd sequence lengths are known for $L \leq 66$ [25] (interestingly, in 1996, the optimal solutions were known for $L \leq 60$ [5], and it took 20 years to prove optimality for six sequences with $61 \leq L \leq 66$). The skew-symmetric solutions are defined only for odd lengths, and the optimal skew-symmetric solutions are known for $L \leq 119$ [25] (previously known results were dated in 2013, $L \leq 89$ [29]).

Theoretical considerations from Golay in 1982 [30] give an upper bound on $F(S)$ of approximately 12.3248 as $L \to \infty$. However, Golay does not prove that 12.3248 is an upper bound on the asymptotic merit factor because it relies on an unproven heuristic argument.

Following the theoretical minimum energy level analysis, a new asymptotic merit factor value of 10.23 was *estimated* by Ukil [31] based on sequences of length 4 to 60, found by exhaustive search.

Heuristic searches among skew-symmetric sequences up to $L = 199$ suggest $F \approx 6$ for long skew-symmetric sequences [32], a value consistent with the results from simulated annealing [6]. Several sequences and merit factor values were reported by Knauer [33], and these values are greater than 6 and lower than 9 for L around 200. A memetic algorithm combined with tabu search was proposed by Gallardo *et al.* [18]. This stochastic algorithm has the efficient implementation of the energy calculation. Borwein *et al.* [15] introduced the directed stochastic algorithm, which was able to find some sequences for length L within the range 149 and 189 with merit factors $F > 9$, and most of them are skew-symmetric. Solver lssOrel [20] was able to find several skew-symmetric sequences with merit factors $F > 8$ for L up to 259.

In 2008, Borwein *et al.* [15] stated in conclusion "We have found good evidence that the upper limit for $\max_L F > 8$ and even > 8.5. These maximal values may routinely exceed 9 in lengths over 200, but it would be difficult to establish this computationally."

Jedwab *et al.* in [34] gave a personal selection of challenges concerning the Merit Factor Problem, arranged in order of increasing significance. The first challenge is as follows: "*Find a binary sequence S of length $L > 13$ for which $F(S) \geq 10$.*"

On the other hand, [35] used the modified Jacobi sequences together with the steep descent, and got an approximate asymptotic merit factor of 6.4382. The gap toward Golay's upper bound, i.e., 12.3248, still remains huge.

We are aware that the study of the merit factor is fundamentally concerned with an asymptotic behavior, and not the identification of a particular sequence with a large merit factor. We hope that this paper, which provides the several merit factors $F > 9$ for $L > 200$ of skew-symmetric sequences, steps forward in researches on the very challenging LABS problem. A known sequence of high merit factor supplies a good initial bound for branch-and-bound and other exact search methods.

III. THE PROPOSED ALGORITHM

In this section, we will describe our stochastic algorithm for a LABS problem. Let us first define a notation, which is relying on our previous work [20], where the self-avoiding walk was used in the lssOrel solver. For the sake of clarity, a pseudocode of lssOrel is presented in Alg. 1.

Solution S can be represented as a coordinate-value pair: $s_i \in \{+1, -1\}$ is represented by 1 and 0, then the coordinate is a binary string of length L, and the value is energy $E(S)$, calculated using (2).

The neighborhood of solution S with length L is obtained by flipping exactly one symbol s_i in the sequence. A neighborhood is required in order to perform a local search or self-avoiding walk. For example, in each step, the gradient walk (steepest descent) moves in a direction to the lowest energy neighbor.

The proposed algorithm, called xLastovka, is presented in Alg. 2. The algorithm starts with an initial solution that is generated randomly. Our algorithm stores a huge number of promising solutions, and performs local searches on these promising solutions in a systematic way.

In Step 4 (Alg. 2), the energy calculation (2) is applied, and stopping criterion ('if target value of energy is reached') is checked in Steps 5–7. Note that the stopping criteria can also be runtime limit, a number of function evaluations *cntProbe* (i.e., energy calculations), etc.

The main while loop is presented in Steps 9–26. The best coordinate, say *pivot*, is obtained from the priority queue *PQ* in Step 10. The pivot's neighborhood is searched inside the for loop, using efficient implementation of the energy calculation [18], [20]. In Step 11, $\frac{L+1}{2}$ flips are required since one skew-symmetric sequence is searched. The while loop stops if the found energy value is better than or equal to the target value, which is an algorithm's parameter.

The xLastovka algorithm uses a priority queue, in which neighbor coordinates are stored, along with their values of energy. Since the coordinate with the best value is the most promising to become the next pivot, the priority queue needs to be ordered. There arises a question about the size of a priority queue. Is it necessary to store all neighbors, or would it be sufficient to store only some of the most promising neighbors? In the latter case, a smaller amount of storage would be required. Therefore, our priority queue stores a predefined number of the best coordinate-value pairs.

Another challenge is regarding memory, and it arises when a solver stores already probed coordinates. The number of the probed coordinates increases very quickly, and it seems impractical to store all visited coordinates. More practical seems to be to store only pivots, but their number also

Algorithm 1 The lssOrel Algorithm [20]

Require: L - instance size
Require: *valueTarget* – best upper bound
Ensure: *coord,value* – best coordinate and best value

```
closePivots ← ∅                          ▷ storage of pivots – hash table
coord ← RandomSolution(L)                ▷ initialize coordinate
value ← Eval(coord)                      ▷ evaluation
if value ≤ valueTarget then              ▷ stopping criteria
    return (coord,value)
end if
while value > valueTarget do
    for k ← 0; k < 8 · (L+1)/2; k++ do   ▷ self-avoiding walk
        iterVal ← ∞
        for i ← 1; i ≤ (L+1)/2; i++ do   ▷ search neighborhood
            S ← coord
            sFlipped ← S.flip(i)         ▷ flip i-th bit
            if closePivots.find(sFlipped) then
                continue                 ▷ skip if already pivot
            else
                value ← Eval(sFlipped)
            end if
            if value ≤ valueTarget then  ▷ stopping criteria
                break
            else if value ≤ iterVal then ▷ random tie if equals
                (iterVal,iterCoord) ← (value,sFlipped)
            end if
        end for
        closePivots.push(coord)
        coord ← iterCoord                ▷ next pivot
    end for
    if value > valueTarget then
        coord ← RandomSolution(L)        ▷ reinitialize coord.
        value ← Eval(coord)              ▷ evaluation
    end if
end while
return (coord,value)
```

Algorithm 2 The xLastovka Algorithm

Require: L - instance size
Require: *valueTarget* – best upper bound
Ensure: *coord,value* – best coordinate and best value

PQ $\leftarrow \emptyset$ ▷ priority queue of pairs (*coord*,*value*)
closePivots $\leftarrow \emptyset$ ▷ storage of pivots – hash table
coord $\leftarrow$ RandomSolution(L) ▷ initialize coordinate
value $\leftarrow$ Eval(*coord*) ▷ evaluation
if *value* $\leq$ *valueTarget* **then** ▷ stopping criteria
    return (*coord*,*value*)
end if
PQ.push(*coord*,*value*)
while *value* $>$ *valueTarget* **do**
    coord $\leftarrow$ PQ.pop()
    for $i \leftarrow 1$; $i \leq \frac{L+1}{2}$; i++ **do** ▷ search neighborhood
        $S \leftarrow$ *coord*
        sFlipped $\leftarrow$ S.flip(i) ▷ flip i-th bit
        if closePivots.find(*sFlipped*) **then**
            continue ▷ skip if already pivot
        else
            value $\leftarrow$ Eval(*sFlipped*)
        end if
        if *value* $\leq$ *valueTarget* **then** ▷ stopping criteria
            break
        else
            PQ.push(*sFlipped*,*value*)
        end if
    end for
    closePivots.push(*coord*)
end while
return (*coord*,*value*)

increases very quickly. It is obvious that a solver needs to store as many as possible coordinates in order to avoid making a cycle, i.e., repeating pivots. The best scenario, when a solver could store all coordinates, is limited by the amount of a computer's memory.

As a mechanism that tries to overcome the memory limitation, we can use tabu search to store symbol s_i, which is flipped, for some steps (iterations) in the optimization process. Then this symbol remains fixed, and it could be flipped again, if necessary, after some steps. This mechanism, which is used in memetic algorithm [18], does not require a big amount of memory. It uses less memory than the lssOrel and xLastovka algorithms, which is a great advantage. However, its disadvantage lies in an issue that it does not prevent strictly a repetition of the pivot(s), which has a consequence in the lower performance of the memetic algorithm combined with tabu search in comparison with the other two algorithms. A detailed analysis of the memetic algorithm combined with tabu search and the lssOrel algorithm is presented in [20]. The memetic algorithm uses less memory than the lssOrel algorithm with the Self-Avoiding Walk (SAW) with the length of $8 \cdot \frac{L+1}{2}$. The lssOrel algorithm stores all pivots of one contiguous SAW, whereas the proposed xLastovka algorithm stores pivots and the predefined number of the best coordinate-value pairs.

In this work, the xLastovka algorithm used the following parameters. The size of priority queue *PQ* was 640 000 coordinates, and the size of storage for pivots was 512 MB. The parameters were set based on some additional experiments, but we did not perform a fine tuning of the parameters.

TABLE 1. Best, worst, median, mean and standard deviation values for *cntProbe* of the xLastovka solver to get the best known skew-symmetric solution.

L	Best	Worst	Median	Mean	StdDev
27	15	4066	874.5	1667	1421
41	162	2.096e+04	3710	4992	4356
43	3642	1.255e+05	6.592e+04	6.961e+04	1.809e+04
45	639	1.246e+05	5.808e+04	5.446e+04	3.399e+04
49	2370	2.505e+05	1.546e+05	1.311e+05	5.779e+04
51	7.429e+05	1.406e+06	1.071e+06	1.084e+06	1.454e+05
53	1.154e+05	1.037e+06	7.586e+05	7.321e+05	1.722e+05
57	4571	4.315e+06	3.276e+06	3.237e+06	6.406e+05
59	3674	5.052e+05	8.981e+04	1.451e+05	1.341e+05
71	6697	3.418e+05	1.39e+05	1.532e+05	8.593e+04
77	2.824e+06	6.073e+07	9.863e+06	1.411e+07	1.232e+07
83	1.74e+05	3.151e+07	9.846e+06	1.142e+07	6e+06
91	2.705e+05	1.311e+08	2.35e+07	3.131e+07	2.686e+07
95	6.431e+06	4.686e+08	8.795e+07	1.108e+08	8.72e+07
97	2.613e+06	1.906e+08	1.973e+07	3.004e+07	3.166e+07
99	7.963e+06	3.155e+08	1.271e+08	1.202e+08	7.551e+07
101	2.562e+06	8.943e+08	1.679e+08	1.852e+08	1.4e+08
103	7.676e+05	1.383e+08	2.381e+07	3.342e+07	2.907e+07
105	6.687e+04	7.066e+08	3.821e+07	7.372e+07	1.085e+08
107	1.398e+09	4.158e+10	7.528e+09	9.856e+09	7.41e+09
109	7.463e+05	8.279e+08	1.323e+08	1.741e+08	1.604e+08
111	6.713e+06	1.337e+09	3.165e+08	3.533e+08	2.537e+08
115	2.1e+08	1.174e+10	3.159e+09	3.625e+09	2.385e+09
117	6.985e+05	4.579e+09	1.292e+09	1.368e+09	9.473e+08
119	5.5e+07	7.383e+09	2.716e+09	2.755e+09	1.497e+09
121	6.288e+05	1.707e+10	6.461e+09	6.548e+09	3.589e+09
123	2.522e+08	2.233e+10	3.479e+09	5.828e+09	5.058e+09
125	3.732e+06	1.49e+11	2.663e+10	3.35e+10	2.691e+10
127	2.655e+06	9.502e+09	3.485e+09	3.474e+09	2.388e+09

IV. RESULTS

In this section, we present experimental results: (1) An analysis of the proposed xLastovka algorithm on small and middle size skew-symmetric sequences, and a comparison with state-of-the-art algorithms, and (2) xLastovka, tested on larger skew-symmetric sequences, where we tried to find new best-known solutions.

A. RESULTS ON SKEW-SYMMETRIC SEQUENCES WHEN $L \leq 127$

In this experiment, we ran the xLastovka algorithm 100 times for skew-symmetric sequences that have 4 optima or one canonic solution [20]. Each run was stopped when a target value of energy was reached. The target value was equal to the currently best known (optimal for $L \leq 119$) energy value of skew-symmetric sequences. The obtained results are shown in Table 1, where the best, worst, mean, median and standard deviation values of *cntProbe* are presented.

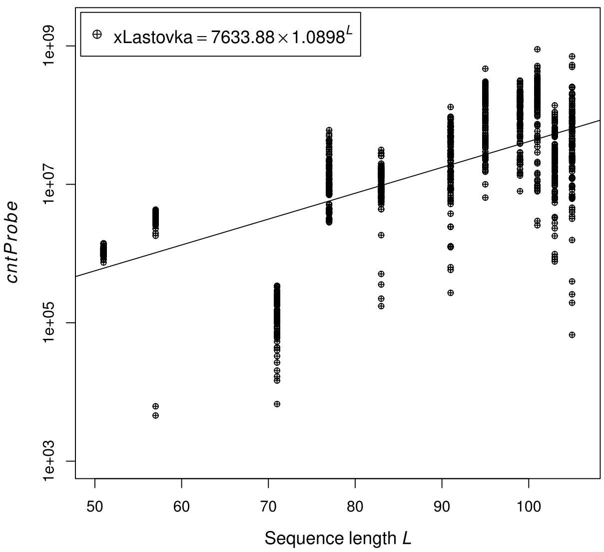

FIGURE 1. The asymptotic performance comparison of xLastovka on lengths from Table 1 for $L = 51$ up to $L = 105$.

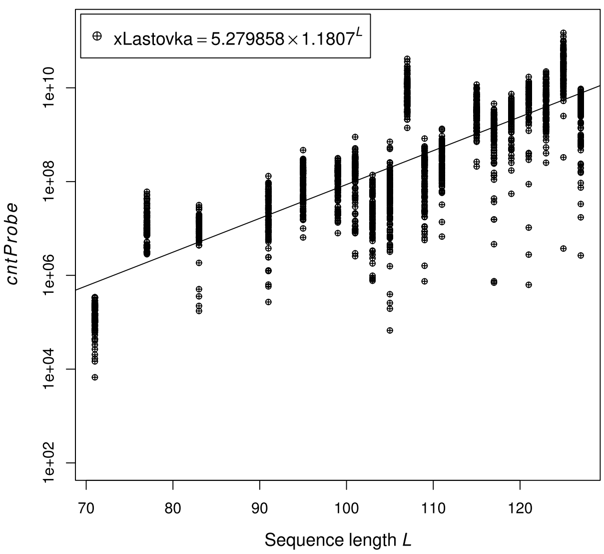

FIGURE 2. The asymptotic performance comparison of xLastovka on lengths from Table 1 for $L = 71$ up to $L = 127$.

Asymptotic performance for L between 51 and 105 is presented in Fig. 1, and for L between 71 and 127 in Fig. 2. For each L, the *cntProbe* does not follow a normal distribution, but it follows an exponential distribution. Therefore, it can be noticed that asymptotic performance may deviate slightly for different sequence length ranges. Note that *cntProbe* is plotted on a logarithmic scale in Figs. 1 and 2. In our previous paper [20], we showed that the time is highly correlated with a number of function evaluations, *cntProbe*. Based on results for L between 71 and 127 (Fig. 2), the estimated complexity of our algorithm is $O(1.2^L)$.

Let us make a comparison of xLastovka with lssOrel [20] regarding speed. The speed is defined as the number of probes per second. The lssOrel solver can reach a higher speed. For example, lssOrel can perform approx. 9.2e+6 probes per second for $L = 127$, while xLastovka reaches approx. 2.9e+6 probes per second. However, xLastovka was able to find some LABS solutions with higher merit factors on larger sequences, as we can see in the next subsection.

B. RESULTS ON LARGE SKEW-SYMMETRIC SEQUENCES

In this section, experiments were conducted on sequences with long length, for $L = 149$ up to $L = 225$. In order to make a comparison with other results from the literature, we divided this interval into two parts: $149 \leq L \leq 189$ and $191 \leq L \leq 225$. The experiments were performed using the SLING grid computing environment [36].

The directed stochastic algorithm [15] was able to find ten solutions with $F > 9$ for $L = 149$ up to $L = 189$, while xLastovka was able to find solutions with the same merit factors, too. Note that the directed stochastic algorithm did not find sequences with merit factor $F > 9$ for L in the range 191–200.

The xLastovka algorithm has found better or equal solutions for L in the range 149–225 than those reported in [33] in all cases except for 159, 167, 187. These exceptions indicate that a particular stochastic algorithm may not perform as the winner in all cases.

The most interesting part of the obtained results ($191 \leq L \leq 225$) is presented in Fig. 3 and Table 2. Figure 3 shows merit factors obtained by the xLastovka algorithm and compared with the results obtained by the solver lssOrel [20] and memetic algorithm [18], and also with the values reported by Knauer [33]. The lssOrel algorithm reported two solutions in the range 191–225, and both skew-symmetric solutions have better merit factors than those reported in [33], while xLastovka improved both of them further.

Best-known solutions obtained by xLastovka for $L = 191$ up to $L = 225$ are summarized in Table 2, where energy, merit factor, and best coordinate are reported. The proposed xLastovka algorithm has found 18 new best-known solutions. Among these solutions, there are three new best-known skew-symmetric solutions for $191 \leq L \leq 199$ with merit factor $F > 9$ (L =191, 193, 199), and 6 new best-known skew-symmetric sequences with merit factor greater than 9 in lengths over 200. These sequences are 201, 207, 211, 213, 223, and 225.

Till now, only two merit factors greater than 9.5 were known, i.e., for lengths 103 and 177. In this work, our algorithm adds two new, $F = 9.5851$ and $F = 9.5393$ for

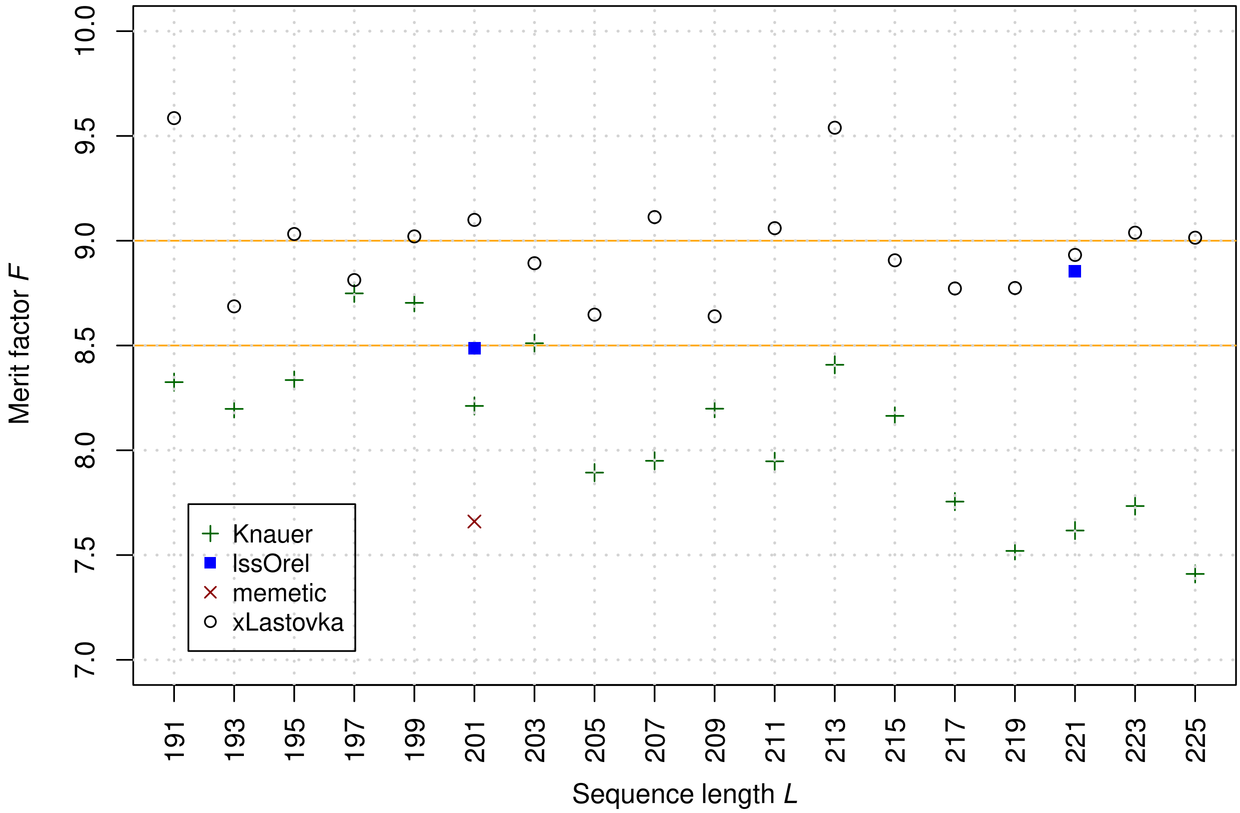

FIGURE 3. Merit factors of binary sequences obtained by the solver xLastovka, memetic algorithm [18], the solver lssOrel [20], and also with the values reported by Knauer [33].

lengths 191 and 213, respectively, and both of them are skew-symmetric.

The obtained results improve the suggestion from [6] and [32] greatly, where $F \approx 6$ for long skew-symmetric sequences up to $L = 199$.

Figure 4 shows best-known merit factors up to $L = 401$ for sequences of odd length. In this Figure, we can see clearly that merit factors drop for a length greater than 225. We believe that our and also other state-of-the-art stochastic algorithms were not able to find sequences with merit factors greater than 9 when $L > 225$. We also believe that such sequences exist for L, which is slightly greater than 225 – for even much larger sequences it is hard to judge.

We make an estimation of merit factors from the range $51 \le L \le 225$ of odd length by a linear model

$$8.6770 + 0.0015 \times L \tag{6}$$

that is also shown in Fig. 4. The red line in Fig. 4 will reach the value $F = 10$ at $L = 883$, which is a big gap, since our model uses values up to 225, and, therefore, it is hard to make an accurate prediction for larger L. This model is very consistent with the model of the lssOrel [20]

$$8.6325 + 0.0007581 \times L, \tag{7}$$

where the estimation is performed on the range $51 \le L \le 183$. The model (6) is calculated using 21 more

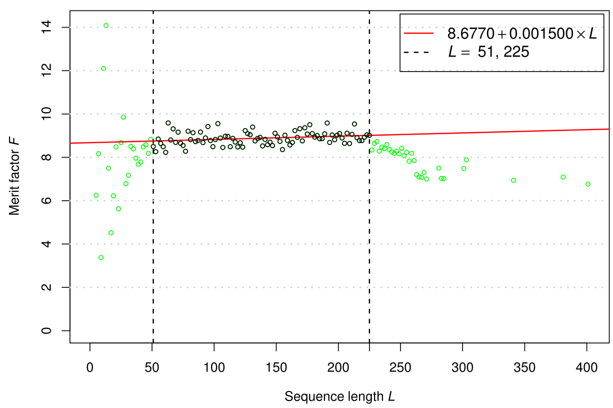

FIGURE 4. Merit factors of binary sequences with odd lengths (optimal for $L \le 119$, best-known otherwise). The merit factors drop when $L > 225$. A linear model is calculated on the range $51 \le L \le 225$.

points with larger lengths ($183 < L \le 225$) on sequences that are skew-symmetric. Note again, that the skew-symmetric solutions might not be optimal for each problem instance.

The greatest merit factor $F = 14.08$ is known for length 13. For smaller lengths, the merit factor values are quite dispersed. It is interesting to look at values of merit factors for lengths between 100 and 225 in Fig. 4. These values are arranged pretty well around 9. Note that

$$F > 8.5, \quad \text{for } 157 < L \le 225 \text{ and } L \text{ is odd} \tag{8}$$

TABLE 2. Best-known merit factors and solutions obtained with the xLastovka solver. The coordinate is presented in the run-length encoding notation.

L	E	F	coordBest
191	1903	**9.5851**	2,2,3,1,1,2,2,3,1,2,4,3,2,4,3,3,1,1,2,2,3,1,2,1,1,1,2,1,4,1,1,1,3,3,1,4,6,7,1,1,1,1,1,1,3,1,8,1,1,1,1,1,2,1,1,1,1,2,1,1,3,1,2,1,5,1,1,3,5,3,1,2,2,4,1,2,1,2,1,1,2,2,1,2,1,1,2,3,1,2,2,4,1,2,2,1
193	2144	8.6868	2,3,4,4,1,3,5,1,6,2,3,1,1,12,4,1,1,1,1,4,4,1,2,2,3,1,1,1,5,2,1,2,3,1,1,2,2,2,2,2,2,4,1,2,3,2,1,1,1,5,1,2,2,3,1,1,2,1,1,6,1,1,2,1,1,1,1,1,1,1,1,1,4,1,2,2,1,1,1,1,3,1,1,1,2,1,3,1,1,2,1,1,2,1,2,1
195	2105	**9.0321**	3,2,2,1,4,2,2,1,3,2,1,1,2,1,2,2,1,1,2,1,2,1,4,2,2,1,3,5,3,1,1,5,1,2,1,3,1,1,2,1,1,1,1,2,1,1,1,1,1,8,1,3,1,1,1,1,1,1,7,6,4,1,3,3,1,1,1,4,1,2,1,1,1,2,1,3,2,2,1,1,3,3,4,2,3,4,2,1,3,2,2,1,1,3,2,2,1,1
197	2202	8.8122	2,1,1,2,1,1,2,1,2,1,1,2,1,2,2,1,2,6,2,1,2,1,1,1,1,1,2,1,1,2,2,1,6,1,2,2,6,1,2,3,1,1,1,1,1,1,3,2,1,5,1,6,1,3,1,1,1,1,3,1,1,1,3,2,1,8,1,2,3,1,1,1,1,2,2,3,1,1,1,1,3,2,4,7,3,2,1,1,1,1,2,3,2,3,4,3,4,4,1
199	2195	**9.0207**	16,1,2,1,2,2,1,6,2,2,2,1,2,2,2,1,2,5,1,1,2,1,1,4,5,1,2,5,2,4,5,2,1,2,2,4,1,1,4,1,1,2,2,3,2,1,1,1,2,1,1,2,2,1,1,1,2,3,1,1,1,2,1,1,4,4,1,1,1,2,3,2,2,3,2,2,2,1,1,1,1,3,2,3,3,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
201	2220	**9.0993**	5,6,3,5,7,1,2,2,3,1,1,3,1,6,2,1,3,1,1,1,3,2,3,2,4,2,2,1,2,4,3,3,1,1,1,4,1,2,5,1,1,1,2,3,1,1,5,1,2,1,2,1,1,2,3,2,2,1,1,2,2,1,2,2,1,5,1,3,2,1,1,1,1,3,1,4,1,2,2,3,1,1,1,1,1,2,1,1,1,2,1,2,1,1,1,1,2,1,1,1,1
203	2317	8.8927	3,3,3,4,3,3,3,4,6,4,5,3,1,3,8,3,2,2,9,1,1,1,3,1,2,2,2,1,1,1,1,3,3,1,3,2,2,2,1,3,1,2,1,6,2,2,3,1,5,1,1,1,1,1,1,1,2,2,2,1,2,1,1,1,1,1,1,2,1,3,1,2,1,1,1,2,1,1,2,1,1,1,1,2,1,1,2,1,2,1,2,1,2,1,1,2,1,2,1,2,1,1
205	2430	8.6471	2,4,3,4,4,3,1,5,1,1,1,3,2,1,1,6,4,1,2,1,3,2,1,1,1,1,2,1,3,1,1,3,1,2,2,2,1,1,1,1,1,1,1,3,1,2,1,6,1,2,1,2,3,3,1,1,1,1,3,3,1,9,2,2,3,1,4,1,3,6,2,1,3,3,1,1,2,1,1,1,1,4,2,1,5,1,1,1,3,1,2,1,1,2,1,1,2,1,2,1,1,2,1
207	2351	**9.1129**	4,1,2,1,2,1,2,1,2,1,8,1,4,2,2,1,2,4,2,1,2,3,1,2,1,1,2,1,2,3,1,2,1,1,1,1,4,1,6,6,1,3,1,2,1,1,1,1,4,1,2,3,1,1,6,3,1,3,1,1,1,1,2,1,1,1,1,3,1,1,6,3,1,2,3,4,3,1,2,3,2,1,1,2,3,2,2,1,1,3,1,1,1,1,1,1,3,3,3,3,3,1,1,1
209	2528	8.6394	3,1,1,1,1,1,4,3,3,1,1,1,1,1,1,1,2,3,3,2,3,3,5,3,3,1,1,2,3,1,2,1,1,1,1,1,4,1,1,1,2,2,5,2,1,2,1,1,1,1,1,3,2,1,2,2,2,2,2,3,2,1,7,3,2,1,1,1,2,2,5,1,1,7,3,1,2,4,1,2,1,2,1,1,1,2,1,2,1,2,2,1,2,1,2,9,1,2,1,2,1,1,7,1,1
211	2457	**9.0600**	3,3,3,3,3,3,5,3,1,5,3,1,3,1,1,1,3,2,2,5,2,2,2,2,6,7,5,1,1,1,1,1,4,2,6,5,1,1,1,5,1,1,1,2,1,1,1,1,2,2,1,1,7,1,1,1,2,1,1,1,1,1,2,1,1,1,1,2,2,2,2,2,1,1,1,2,2,2,1,5,1,3,1,2,1,1,1,3,1,2,1,1,1,2,1,2,1,2,1,2,1,2,1,2,1,1
213	2378	**9.5393**	2,1,1,1,1,1,3,1,1,5,1,3,1,1,1,1,1,1,2,1,7,2,1,2,2,4,1,1,1,1,1,2,1,1,2,1,2,4,2,1,3,2,1,2,3,1,2,2,1,2,1,2,2,1,1,2,3,4,3,1,2,1,1,2,1,2,4,2,3,3,2,3,1,2,3,2,1,3,2,1,1,2,3,4,7,1,1,2,2,3,2,1,1,1,1,1,3,8,1,3,1,1,1,4,1,7,1
215	2595	8.9066	5,5,4,2,1,2,4,5,1,2,2,1,5,1,4,2,1,5,1,1,3,9,6,2,3,1,1,1,5,2,1,2,2,2,5,1,3,1,3,2,1,3,1,3,1,1,1,2,2,2,3,2,1,1,1,5,1,2,2,1,1,1,1,2,1,1,1,1,1,1,1,2,1,4,1,1,1,3,2,1,1,3,1,1,1,3,2,3,1,1,1,2,1,1,2,3,2,1,1,2,1,1,1,2,1,1,1,1
217	2684	8.7722	9,2,1,1,2,1,8,6,4,1,4,1,4,2,2,1,1,2,1,2,1,1,1,2,1,3,1,5,3,1,2,3,5,2,1,2,2,2,2,2,5,1,1,1,2,2,2,5,1,1,1,2,2,2,2,2,3,2,1,1,1,2,1,2,3,1,2,1,1,1,3,1,3,5,3,4,2,2,1,1,3,1,1,3,1,1,2,1,1,1,1,2,1,1,1,1,1,1,3,4,2,1,1,1,1,1,1,1,1
219	2733	8.7744	2,1,2,2,1,2,1,2,1,2,1,2,1,2,2,1,2,5,5,3,1,2,1,5,6,2,1,2,2,1,1,7,2,1,2,4,1,1,8,1,1,2,1,1,1,1,1,3,1,3,1,3,1,3,1,3,1,7,4,1,1,1,1,1,1,4,1,1,2,3,2,1,1,1,1,1,4,2,3,2,1,1,1,1,2,1,1,1,3,3,1,2,1,1,1,2,1,1,1,2,3,2,3,3,3,3,3,2,3,1
221	2734	8.9322	6,6,5,2,1,4,2,1,2,2,5,2,1,4,1,2,2,3,2,5,1,2,2,1,3,3,3,1,1,2,2,1,1,2,1,1,1,1,2,2,1,2,1,1,12,1,2,3,1,1,1,1,1,1,1,1,1,1,4,3,2,6,4,2,4,1,2,1,2,1,3,2,3,1,1,1,2,2,1,2,2,3,1,1,3,2,1,1,1,2,2,3,2,1,1,3,2,1,1,1,2,1,1,1,1,2,1,1,1,1,1
223	2751	**9.0383**	5,5,3,5,4,5,1,1,3,3,1,5,7,1,2,9,2,2,3,4,2,2,3,2,1,2,1,3,3,4,2,1,8,2,3,1,3,1,2,2,1,1,1,1,1,1,3,2,1,1,2,1,2,1,3,3,2,1,2,2,2,1,1,2,1,2,2,2,1,1,1,1,1,1,1,2,3,1,1,1,1,1,2,1,1,1,3,1,2,1,4,1,1,1,2,1,1,2,1,1,1,2,1,2,1,1,1,2,1,1,1,1
225	2808	**9.0144**	6,3,6,2,6,2,7,4,1,8,1,4,2,3,8,4,1,3,3,2,1,3,2,3,2,2,1,1,3,4,1,1,1,2,3,3,3,2,1,2,1,2,1,2,5,1,1,2,1,4,2,2,1,2,2,1,3,2,1,2,1,3,1,1,2,1,1,1,1,1,1,2,1,2,2,1,1,3,1,1,1,1,1,1,3,1,1,2,1,1,1,1,1,2,2,1,1,1,1,2,2,1,1,1,1,2,1,2,1,1,1,1,1

is obtained with xLastovka, and also with other state-of-the-art heuristic algorithms. We are aware that the study of the merit factor is fundamentally concerned with an asymptotic behavior, and not the identification of a particular sequence with a large merit factor. However, (8) holds for the contiguous range.

V. CONCLUSION

The main contributions of this papers are: (1) On the interval from 159 up to 225 for odd lengths, all the best-known merit factors are greater than 8.5. (2) Now, the largest best-known skew-symmetric sequence with merit factor greater than 9 has the length 225. (3) A new stochastic algorithm is proposed for solving a Low-Autocorrelation Binary Sequence (LABS) problem.

The proposed xLastovka algorithm was very efficient for large instance sizes. It was able to find 6 new best solutions and their merit factors $F > 9$ for $L = 201$ up to $L = 225$. Two large merit factors $F = 9.5851$ and $F = 9.5393$ were obtained for lengths 191 and 213, respectively, and both of them belong to skew-symmetric sequences. Now, it holds that $F > 8.5$ on the interval of odd lengths $157 < L \leq 225$.

The advantage of the proposed algorithm seems to lie in its usage of a priority queue that stores promising solutions (pivots). The queue enables the algorithm to perform local searches on these solutions systematically.

There might be further possible improvements to the xLastovka algorithm. We can include contiguous self-avoiding walks of some length in the algorithm alongside using a priority queue for storing the promising neighbors. Skew-symmetric sequences have odd length, and they could be used as good initial points for an algorithm when searching for sequences without skew-symmetric restriction or nearest odd length sequences.

ACKNOWLEDGMENT

The authors would like to thank Prof. F. Brglez for helpful discussions. The authors would also like to acknowledge the Slovenian Initiative for National Grid (SLING) for using a computation power for performing the experiments in this paper.

REFERENCES

[1] P. Borwein, K. K. S. Choi, and J. Jedwab, "Binary sequences with merit factor greater than 6.34," *IEEE Trans. Inf. Theory*, vol. 50, no. 12, pp. 3234–3249, Dec. 2004.

[2] A. Ukil, "Low autocorrelation binary sequences: Number theory-based analysis for minimum energy level, barker codes," *Digit. Signal Process.*, vol. 20, no. 2, pp. 483–495, Mar. 2010.

[3] J. M. Velazquez-Gutierrez and C. Vargas-Rosales, "Sequence sets in wireless communication systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1225–1248, 2nd Quart., 2017.

[4] L. Zhao, J. Song, P. Babu, and D. P. Palomar, "A unified framework for low autocorrelation sequence design via majorization–minimization," *IEEE Trans. Signal Process.*, vol. 65, no. 2, pp. 438–453, Jan. 2017.

[5] S. Mertens, "Exhaustive search for low-autocorrelation binary sequences," *J. Phys. A, Math. Gen.*, vol. 29, no. 18, pp. 473–481, 1996.

[6] J. Bernasconi, "Low autocorrelation binary sequences: Statistical mechanics and configuration space analysis," *J. Physsique*, vol. 48, pp. 559–567, Apr. 1987.

[7] 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, no. 1, p. 012006, 2015. [Online]. Available: http://stacks.iop.org/1742-6596/613/i=1/a=012006

[8] 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.

[9] J. Jedwab, D. J. Katz, and K.-U. Schmidt, "Littlewood polynomials with small L^4 norm," *Adv. Math.*, vol. 241, pp. 127–136, Jul. 2013.

[10] J. Jedwab, D. J. Katz, and K.-U. Schmidt, "Advances in the merit factor problem for binary sequences," *J. Combinat. Theory A*, vol. 120, no. 4, pp. 882–906, 2013.

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

[12] H.-G. Jeon, J.-Y. Lee, Y. Han, S. J. Kim, and I. So Kweon, "Fluttering pattern generation using modified legendre sequence for coded exposure imaging," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1001–1008.

[13] H.-G. Jeon, J.-Y. Lee, Y. Han, S. J. Kim, and I. S. Kweon, "Generating fluttering patterns with low autocorrelation for coded exposure imaging," *Int. J. Comput. Vis.*, vol. 123, no. 2, pp. 269–286, 2017.

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

[15] P. Borwein, R. Ferguson, and J. Knauer, "The merit factor problem," *London Math. Soc. Lecture Note Ser.*, vol. 352, p. 52, Jan. 2008.

[16] G. Muhammad, M. Alsulaiman, Z. Ali, K. Malki, T. Mesallam, and M. Farahat, "Voice pathology detection and classification using autocorrelation and entropy features in different frequency regions," *IEEE Access*, to be published, doi: 10.1109/ACCESS.2017.2696056.

[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] 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.

[19] S. Halim, R. H. Yap, and F. Halim, "Engineering stochastic local search for the low autocorrelation binary sequence problem," in *Proc. 14th Int. Conf. Principles Pract. Constraint Programm. (CP)*, 2008, pp. 640–645.

[20] 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.

[21] 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.

[22] 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.

[23] J. Jedwab and K. Yoshida, "The peak sidelobe level of families of binary sequences," *IEEE Trans. Inf. Theory*, vol. 52, no. 5, pp. 2247–2254, May 2006.

[24] C. D. Groot, D. Würtz, and K. H. Hoffmann, "Low autocorrelation binary sequences: Exact enumeration and optimization by evolutionary strategies," *Optim., J. Math. Program. Oper. Res.*, vol. 23, no. 4, pp. 369–384, 1992.

[25] T. Packebusch and S. Mertens, "Low autocorrelation binary sequences," *J. Phys. A, Math. Theor.*, vol. 49, no. 16, p. 165001, 2016.

[26] B. Militzer, M. Zamparelli, and D. Beule, "Evolutionary search for low autocorrelated binary sequences," *IEEE Trans. Evol. Comput.*, vol. 2, no. 1, pp. 34–39, Apr. 1998.

[27] A. Reinholz, "Ein paralleler genetischer algorithmus zur optimierung der binären autokorrelations-funktion," M.S. thesis, Rheinische Friedrich-Wilhelms-Univ. Bonn, Bonn, Germany, Oct. 1993.

[28] D. Żurek, K. Piętak, M. Pietroń, and M. Kisiel-Dorohinicki, "Toward hybrid platform for evolutionary computations of hard discrete problems," *Procedia Comput. Sci.*, vol. 108, pp. 877–886, Dec. 2017.

[29] S. D. Prestwich. (2013). "Improved branch-and-bound for low autocorrelation binary sequences." [Online]. Available: https://arxiv.org/abs/1305.6187

[30] M. J. E. Golay, "The merit factor of long low autocorrelation binary sequences (Corresp.)," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 3, pp. 543–549, May 1982.

[31] A. Ukil, "On asymptotic merit factor of low autocorrelation binary sequences," in *Proc. 41st Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Nov. 2015, pp. 004738–004741.

[32] G. F. M. Beenker, T. A. C. M. Claasen, and P. W. C. Hermens, "Binary sequences with a maximally flat amplitude spectrum," *Philips J. Res.*, vol. 40, no. 5, pp. 289–304, 1985.

[33] LABS. (2016). *LABS Problem: 2002 Merit Factor Records Posted by Knauer*. [Online]. Available: https://github.com/borkob/git_labs

[34] J. Jedwab, "A survey of the merit factor problem for binary sequences," in *Proc. SETA*, 2004, pp. 30–55.

[35] J. M. Baden, "Efficient optimization of the merit factor of long binary sequences," *IEEE Trans. Inf. Theory*, vol. 57, no. 12, pp. 8084–8094, Dec. 2011.

[36] SLING. (Oct. 2017). *SLING—Slovenian Initiative for National Grid*. [Online]. Available: http://www.sling.si/

JANEZ BREST (M'02–SM'17) 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.

He has been with the Laboratory for Computer Architecture and Programming Languages, University of Maribor, since 1993. He is currently a Full Professor and the Head of the Laboratory for Computer Architecture and Programming Languages. 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 an Associate Editor of *Swarm and Evolutionary Computation*.

BORKO BOŠKOVIĆ (M'12) received the B.Sc. and Ph.D. degrees in computer science from the University of Maribor, Maribor, Slovenia, in 2004 and 2010, respectively.

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