IOP Publishing Journal of Physics A: Mathematical and Theoretical
J. Phys. A: Math. Theor. 49 (2016) 165001 (18pp) doi:10.1088/1751-8113/49/16/165001

Low autocorrelation binary sequences

Tom Packebusch[1] **and Stephan Mertens**[1,2]

[1] Inst. f. Theo. Physik, Otto-von-Guericke Universität, PF 4120, D-39016 Magdeburg, Germany
[2] Santa Fe Institute, 1399 Hyde Park Rd, Santa Fe, NM 87501, USA

E-mail: mertens@ovgu.de

Received 9 December 2015, revised 3 February 2016
Accepted for publication 9 February 2016
Published 8 March 2016

Abstract
Binary sequences with minimal autocorrelations have applications in communication engineering, mathematics and computer science. In statistical physics they appear as groundstates of the Bernasconi model. Finding these sequences is a notoriously hard problem, that so far can be solved only by exhaustive search. We review recent algorithms and present a new algorithm that finds optimal sequences of length N in time $O(N\ 1.73^N)$. We computed all optimal sequences for $N \leqslant 66$ and all optimal skewsymmetric sequences for $N \leqslant 119$.

Keywords: Bernasconi model, LABS, Barker sequence, merit factor

1. Introduction

Consider a sequence $S = (s_1, \ldots, s_N)$ with $s_i = \pm 1$. The autocorrelations of S are defined as

$$C_k(S) = \sum_{i=1}^{N-k} s_i s_{i+k} \tag{1}$$

for $k = 0, 1, \ldots, N - 1$, and the 'energy' of S is defined as the sum of the squares of all off-peak correlations

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

The *low-autocorrelation binary sequence* (LABS) problem is to find a sequence S of given length N that minimizes $E(S)$ or, equivalently, maximizes the *merit factor*

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

The LABS problem arises in practical applications in communications engineering, where low autocorrelation sequences are used for example as modulation pulses in radar and sonar ranging [1–3]. A particularly exciting application is the interplanetary radar measurement of spacetime curvature [4].

In mathematics, the LABS problem appears in terms of the Littlewood problem [5, 6], the problem of constructing polynomials with coefficients ± 1 that are 'flat' on the unit circle in the complex plane.

In statistical physics, $E(S)/N$ can be interpreted as the energy of N interacting Ising spins $s_i = \pm 1$. This is the Bernasconi model [7]. It has long-range four-spin interactions and is completely deterministic, i.e. there is no explicit or quenched disorder like in spin-glasses. Nevertheless the ground states are highly disordered—quasi by definition. This self-induced disorder resembles very much the situation in real glasses. In fact, the Bernasconi-model exhibits features of a glass transition like a jump in the specific heat and slow dynamics and aging [8]. A clever variation of the replica method allows an analytical treatment of the Bernasconi model in the high-temperature regime [9, 10]. For the low-temperature regime, analytical results are rare—especially the ground states are not known. Due to this connection to physics we refer to the s_i as spins throughout the paper.

These examples illustrate the importance of the LABS problem in various fields. For more applications and the history of the problem we refer to existing surveys [11, 12]. In this contribution we focus on algorithms to solve the LABS problem. But before we discuss algorithms, we will give a brief survey on what is known about solutions.

2. What is known

The correlation C_k is the sum of $N - k$ terms ± 1, hence the value of $|C_k|$ is bounded from below by

$$|C_k| \geqslant b_k = (N - k) \bmod 2. \tag{4}$$

A binary sequence with $|C_k| = b_k$ is called a Barker sequence [13]. The merit factor of a Barker sequence is

$$F_N^{\text{Barker}} = \begin{cases} N & \text{for } N \text{ even,} \\ \dfrac{N^2}{N-1} & \text{for } N \text{ odd.} \end{cases} \tag{5}$$

If it exists, a Barker sequence is a solution of the LABS problem. Barker sequences exist for $N = 2, 3, 4, 5, 7, 11$ and 13, but probably for no other values of N. In fact it can be proven that there are no Barker sequences for odd values of $N > 13$ [14, 15]. For even values of N, the existence of Barker sequences can be excluded for $4 < N \leqslant 2 \cdot 10^{30}$ [16].

Let F_N denote the maximum merit factor for sequences of length N. It is an open problem to prove (or disprove) that F_N is bounded. For Barker sequences, $F_N \propto N$, and the same is true more generally for sequences such that $|C_k| \leqslant C^\star$ for some constant $C^\star$ that does not depend on N or k. The common belief is that no such sequences exist and that F_N is bounded by some constant.

A non-rigorous argument for F_N being bounded was given by Golay [17]. Assuming that the correlations C_k are independent, he argued that asymptotically $F_N \lesssim 12.3248$, or more precisely, that

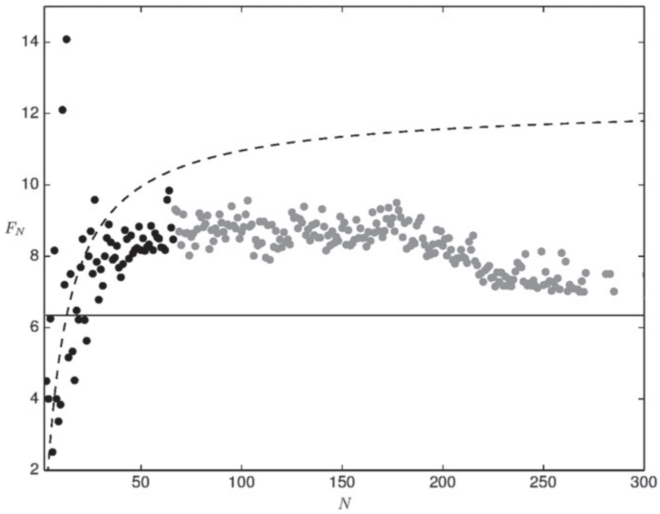

Figure 1. Largest known merit factors. Black symbols are exact solutions from exhaustive searches, gray symbols lower bounds from heuristic or partial searches. The solid line is the rigorous asymptotic lower bound 6.342 061… from appended rotated Legendre sequences [20], the dashed line is Golay's non-rigorous asymptotic upper bound (6). Data from the tables in section 4 and from [21] and references therein.

$$F_N \lesssim \frac{12.3248}{(8\pi N)^{\frac{3}{2N}}}. \tag{6}$$

There are some rigorous results for lower bounds on F_N. The mean value of $1/F$, taken over all binary sequences of length N, is $(N-1)/N$ [18]. Hence we expect $F_N \geqslant 1$. In fact one can explicetly construct sequences for all values of N that have merit factors larger than 1. The current record is set by so called appended rotated Legendre sequences with an asymptotic merit factor of 6.342 061… [19, 20].

Beyond that, our knowledge about solutions of the LABS problem is based on computer searches. Figure 1 shows the best merit factors known for $N < 300$. For small values of N, we can exhaustively search through all sequences to find the sequences with the maximum merit factor F_N. An evaluation of $E(S)$ from scratch takes time $\Theta(N^2)$, but one can loop through all sequences such that any two successive sequences differ by exactly one spin, an arrangement known as Gray code [22]. The corresponding update of $E(S)$ takes only linear time, and the total time complexity of exhaustive enumeration is then given by $\Theta(N\ 2^N)$. In this paper we will discuss a class of exact enumeration algorithms with time complexity $\Theta(N\ b^N)$ with $b < 2$ that we used to solve the LABS problem up to $N \leqslant 66$.

For larger values of N exhaustive enumeration is not feasible and one has to resort to either partial enumerations or heuristic searches. In both cases one obtains sequences with large but not necessarily maximal merit factors.

Partial enumerations are exhaustive enumerations of a well defined subset of sequences. A particular promising subset is given by skewsymmetric sequences of odd length $N = 2n-1$. These sequences satisfy

$$s_{n+\ell} = (-1)^\ell s_{n-\ell} \qquad (\ell = 1, \ldots, n-1), \tag{7}$$

which implies that $C_k = 0$ for all odd k. The restriction to skewsymmetric sequences reduces the size of the search space from 2^N to $2^{N/2}$. Sequences with maximum merit factor are often, but not always skewsymmetric: from the 31 LABS problems for odd $N \leqslant 65$, 21 have skewsymmetric solutions (section 4). For the other values of N, skewsymmetric sequences provide lower bounds for F_N. We used our enumeration algorithm to compute the optimal skewsymmetric sequences for all $N \leqslant 119$.

Enumerative algorithms (complete or partial) are limited to small values of N by the exponential size of the search space. Heuristic algorithms use some plausible rules to locate good sequences more quickly. Examples are simulated annealing, evolutionary algorithms, tabu search—the list of heuristic algorithms that have been applied to the LABS problem is much longer, see [23]. The state of the art are the solvers described in [21], which have found many of the merit factors shown in figure 1. The figure shows a significant drop of the merit factors for $N > 200$. This is generally attributed to the fact that even sophisticated search heuristics fail for LABS problems of larger size. This hardness has earned the LABS problem a place in CSPLIB, a library of test problems for constraint solvers [24, problem 005].

3. Algorithm

According to the current state of knowledge, the only way to get exact solutions for the LABS problem is exhaustive search. With a search space that grows like 2^N, this approach is limited to rather small values of N, however. The exponential complexity calls for a method to restrict the search to smaller subspaces without missing the exact solutions. This is where branch and bound comes in, a powerful and versatile method from combinatorial optimization [25]. All exact solutions of the LABS problem for $N > 32$ have been obtained with variations of a branch and bound algorithm proposed in [26] that reduces the size of the search space from 2^N to b^N with $b < 2$. In this section we review these algorithms and we present a new variant which has $b = 1.72$, the best value to date.

The idea of branch and bound is to solve a discrete optimization problem by breaking up its feasible set into successively smaller subsets (*branch*), calculating bounds on the objective function value over each subset, and using them to discard certain subsets from further consideration (*bound*) [25]. The procedure ends when each subset has either produced a feasible solution, or has been shown to contain no better solution than the one already in hand. The best solution found during this procedure is a global optimum.

The goal is of course to discard many subsets as early as possible during the branching process, i.e. to discard most of the feasible solutions before actually evaluating them. The success of this approach depends on the branching rule and very much on the quality of the bound, but it can be quite substantial.

For the LABS problem we specify a set of feasible solutions be fixing the m leftmost and the m rightmost spins of the sequence. The $N - 2m$ centre spins are not specified, i.e. the set contains 2^{N-2m} feasible solutions. Given a feasible set specified by the $2m$ outer elements, four smaller sets are created by fixing the elements s_{m+1} and s_{N-m} to ± 1 and m is increased by 1. This is applied recursively until all elements have been fixed. This is the branching rule introduced by the original branch and bound algorithm [26], and it is shared by all later versions. It has the nice property that the long range correlations are fixed early in the recursion process. Specifically, if the m left- and rightmost spins are fixed, all C_k for $k \geqslant N - m$ are fixed. In addition, this branching rule supports the computation of lower bounds very well, as we will see below.

The branching process can be visualized as a tree in which nodes represent subsets. Each node has four children corresponding to the four possible ways to set the two spins in the $(m + 1)$th shell. The branch and bound algorithm traverses this tree and tries to exclude as many branches as possible by computing a bound on the energy that can be achieved in a branch. The number of nodes actually visited is a measure of quality for the bound.

3.1. Bounds

Bounds are usually obtained by replacing the original problem over a given subset with an easier (relaxed) problem such that the solution value of the latter bounds that of the former. A good relaxation is one that is easy and fast to solve and yields strong lower bounds. Most often these are conflicting goals.

An obvious relaxation of the LABS problem is given by the problem to minimize all values C_k^2 *independently*. Hence we replace the original problem

$$E_{\min} = \min_{\text{free}} \left(\sum_{k=1}^{N-1} C_k^2 \right) \tag{8}$$

by the relaxed version

$$E_{\min}^* = \sum_{k=1}^{N-1} \min_{\text{free}} (C_k^2) = \sum_{k=1}^{N-1} (\min_{\text{free}} (|C_k|))^2 \leqslant E_{\min}, \tag{9}$$

where 'free' refers to the $N - 2m$ center elements of s that have not yet been assigned. All previous branch and bound approaches to LABS considered $E_{\min}^*$ to be too expensive to compute and replaced it by a weaker, but easily computable bound $E_b \leqslant E_{\min}^*$ obtained from bounding $\min_{\text{free}} |C_k|$ from below.

3.1.1. The original bound. In the original algorithm [26] the bound E_b is computed by assigning (arbitrary) values to all free spins, thereby fixing the values for all correlations to $C_k^\star$. Since flipping a free spin can decrement $|C_k|$ at most by 2, a lower bound for $|C_k|$ is given by

$$\min_{\text{free}} |C_k| \geqslant \max(b_k, |C_k^\star| - 2\hat{f}_k), \tag{10}$$

where

$$\hat{f}_k = \begin{cases} 0 & \text{if } k \geqslant N - m, \\ 2(N - m - k) & \text{if } N/2 \leqslant k < N - m \text{ or} \\ N - 2m & \text{otherwise} \end{cases} \tag{11}$$

denotes the number of free spins that appear in C_k and b_k is given by (4). The running time of this algorithm scales like $O(1.85^N)$. A parallelized version of the algorithm was used to solve the LABS problem up to $N = 60$ [27].

3.1.2. The Prestwich bound. The quality of the bound (10) depends on the values of $C_k^\star$ and hence on the arbitrary values assigned to the free spins. In principle, these values should be chosen to maximize $C_k^\star$, but this requires the solution of another optimization problem for each bound. This can be avoided by considering free *products* instead of free spins: a product $s_i s_{i+k}$ is free if s_i or s_{i+k} is a free spin. Products $s_i s_{i+k}$ in which both spins are fixed ars called fixed. Let $c_k(s)$ denote the sum of all fixed products that contribute to C_k. Note that $c_k = C_k$ for $k \geqslant N - m$. Then

$$\min_{\text{free}} |C_k| \geqslant \max(b_k, |c_k(s)| - f_k), \tag{12}$$

where

$$f_k = (N - k) - 2\max(m - k, 0) - \max(k - N + 2m, 0) \tag{13}$$

denotes the number of free products in C_k, and b_k is given by (4). The reasoning behind (12) is that the sum c_k of fixed products may be offset by the sum of free products, which is no greater than f_k. If $|c_k(s)| > f_k$ then $|c_k(s)| - f_k$ is a lower bound for $|C_k|$. If $|c_k(s)| \leqslant f_k$, this bound is useless and we have to resort to the trivial lower bound $|C_k(s)| \geqslant b_k$. The bound (12) was used by Prestwich to prune parts of the search space in a local search algorithm for the LABS problem [28].

For his recent branch and bound algorithm for LABS, Prestwich [29] improved that bound by taking into account some of the interactions between fixed and free spins. Suppose that s_i is a free spin while s_{i-k} and s_{i+k} are fixed. If $s_{i-k} \neq s_{i+k}$, the contributions

$$s_{i-k}s_i + s_i s_{i+k} = s_i(s_{i-k} + s_{i+k}) \tag{14}$$

of s_i to C_k are zero, no matter what the value of s_i is. For each such *cancellation*, the number f_k in (12) can be decreased by two. For $s_{i-k} = s_{i+k}$, the contribution of the term (14) is ± 2, a situation referred to as *reinforcement* by Prestwich. Now, if all free contributions to C_k are either cancellations or reinforcements, then f_k must be even. If the sum of the fixed contributions c_k is also even and $c_k \bmod 4 \neq f_k \bmod 4$, we can set $b_k = 2$ in (12). With this bound, Prestwich reports a running time that scales like $O(1.80^N)$. Since Prestwich did not parallelize his algorithm, this estimate was based on enumerations only up to $N \leqslant 44$.

3.1.3. The Wiggenbrock bound. A different bound was used by Wiggenbrock in his branch and bound algorithm [30]. Flipping a spin changes the sum $C_k + C_{N-k}$ by ± 4 because every spin occurs twice in that sum. Taking the all $+1$ configuration as a reference, we get

$$(N - C_{N-k}) \equiv C_k \pmod{4}. \tag{15}$$

For $k \geqslant N - m$, the C_k are completely fixed. For other values of k, the correlations can be bounded by

$$|C_k| \geqslant \begin{cases} |(N - C_{N-k}) \bmod 4| & \text{if } k \leqslant m, \\ b_k & \text{if } m < k < N - m, \end{cases} \tag{16}$$

where we assumed the residue system $\{-1, 0, 1, 2\}$ for the mod 4 operation.

The Wiggenbrock bound seems to be weak since it bounds $|C_k|$ by small numbers 0, 1, 2 only. Yet it is surprisingly efficient: Wiggenbrock reported a running time of $O(1.79^N)$, slightly better than the scaling of Prestwich's bound. Using a parallelized implementation and running it on 18 GPUs, Wiggenbrock solved the LABS problem for $N \leqslant 64$ [30].

3.1.4. The combined bound. High up in the search tree, where m is small, the contributions of the free products overcompensate the fixed contributions and the Prestwich bound (12) reduces to b_k. The Wiggenbrock bound (16) provides a better bound in exactly these situations. The fact that it yields such a good running time indicates that even this weak bound is efficient because it applies high up in the search tree: a branch, that can be pruned at this level, is usually very large. The Prestwich bound with the free products applies for larger values of m, on the other hand. An obvious idea is to combine these complimentary bounds and use

$$|C_k| \geqslant \begin{cases} \max(|(N - C_{N-k}) \bmod 4|, |c_k(s)| - f_k) & \text{if } k \leqslant m, \\ \max(b_k, |c_k(s)| - f_k) & \text{if } m < k < N - m, \end{cases} \tag{17}$$

as a bound.

When we measure the number of recursive calls (i.e. the number of nodes visited in the search tree) and the CPU time per call (figure 2), we find that the running time of the branch and bound algorithm with the combined bound (17) scales like $\Theta(N 1.729^N)$.

3.1.5. The tight bound. So far, all bounds are lower bounds for $\min|C_k|$ that yield only a lower bound for $E^*_{\min}$, which already is a lower bound for $E_{\min}$. We will now show that $\min|C_k|$ and hence $E^*_{\min}$ can be computed exactly, thereby avoiding the 'second relaxation' to E_b and providing the best lower bound possible from the ansatz (9). We write C_k as

$$C_k = c_k + u_k, \tag{18}$$

where c_k is the sum of all fixed terms $s_i s_{i+k}$ (as above) and u_k sums up all terms in which at least one spin is free. Let

$$g_k = \begin{cases} 4 & \text{if } k \leqslant m, \\ 2 & \text{otherwise.} \end{cases} \tag{19}$$

We will show below that there exist easy to compute integers $U_k^{\min}$ and $U_k^{\max}$ such that the free contribution u_k can take on all values in

$$\{U_k^{\min}, U_k^{\min} + g_k, U_k^{\min} + 2g_k, \ldots, U_k^{\max} - g_k, U_k^{\max}\}. \tag{20}$$

All we need to know are the values of c_k, $U_k^{\min}$ and $U_k^{\max}$ to compute

$$\min|C_k| = \begin{cases} c_k + U_k^{\min} & \text{if } -c_k \leqslant U_k^{\min}, \\ c_k + U_k^{\max} & \text{if } -c_k \geqslant U_k^{\max}, \\ |(-c_k - U_k^{\min}) \bmod g_k| & \text{otherwise,} \end{cases} \tag{21}$$

and then $E^*_{\min} = \sum_k (\min|C_k|)^2$.

To prove (19) and (20), we rearrange the sum (1) for C_k a little bit. For C_3 and $N = 12$, for example, we can write

$$\begin{aligned} C_3 &= s_1 s_4 + s_1 s_4 + s_1 s_4 + s_1 s_4 + s_1 s_4 + s_1 s_4 + s_1 s_4 + s_1 s_4 + s_1 s_4 \\ &= (s_1 s_4 + s_4 s_7 + s_7 s_{10}) + (s_2 s_5 + s_5 s_8 + s_8 s_{11}) + (s_3 s_6 + s_6 s_9 + s_9 s_{12}). \end{aligned}$$

We call every sum in parentheses a *chain*. For general values of k and N we write

$$C_k = \sum_{j=1}^{k} \sum_{q=1}^{\left\lfloor \frac{N-j}{k} \right\rfloor} s_{j+(q-1)k} s_{j+qk}. \tag{22}$$

The chains are the sums over q. For $k < N - m$, each chain contains a subchain of free terms

$$s_a s_{a+k} + s_{a+k} s_{a+2k} + \cdots + s_{b-k} s_b, \tag{23}$$

where only the spins s_a and s_b may be fixed. We refer to these subchains as free chains. The sum of all free chains equals u_k.

Let us first prove the 'granularity' (19). If both spins s_a and s_b are fixed, then every free spin appears exactly in two terms, and flipping any free spin changes the sum (23) by 0 or ± 4. If either s_a or s_b (or both) are free, then flipping this spin changes the sum (23) by ± 2. Hence the granularity g_k is 4 if and only if all contributing free chains have both s_a and s_b fixed, and 2 otherwise.

Now s_a can be free and the leftmost member of a free chain if and only if $a > m$ and if it has no left partner, i.e. if $a - k \leqslant 0$. Together, both conditions imply $k > m$. Hence by argumentum e contrario, $k \leqslant m$ implies that s_a is fixed and, by similar reasoning, also that that s_b is fixed. This proves that $g_k = 4$ for $k \leqslant m$.

If $k > m$, we only need to find a single free chain that starts with a free spin. Consider the spin s_{m+1}: it is free and it has no left neighbor. Hence it is the leftmost spin of a free chain that contributes to u_k. Therefore $g_k = 2$ for $k > m$. Note that for $k > m$ there can be free chains with both s_a and s_b fixed. All we have proven is that for $k > m$ this can not happen for all free chains.

Now we will prove (20). Let n denote the number of terms $s_j s_{j+k}$ in a free chain (23), and let u denote its value. If s_a or s_b (or both) are free, then u can take on all values between $-n$ and n with granularity 2:

$$u \in [-n, -n+2, \ldots, n-2, n] \qquad (s_a \text{ or } s_b \text{ free}). \tag{24}$$

If both spins s_a and s_b are fixed, the granularity is 4 and the range of values varies with s_a, s_b and the parity of n according to

$$u \in \begin{cases} [-n, \ldots, n] & \text{if } s_a = s_b \text{ and } n \text{ even,} \\ [-(n-2), \ldots, n] & \text{if } s_a = s_b \text{ and } n \text{ odd,} \\ [-(n-2), \ldots, (n-2)] & \text{if } s_a \neq s_b \text{ and } n \text{ even,} \\ [-n, \ldots, (n-2)] & \text{if } s_a \neq s_b \text{ and } n \text{ odd.} \end{cases} \tag{25}$$

This can be proven by induction over n. For n odd, the base case is $n = 3$, i.e.

$$u = s_a s_{a+k} + s_{a+k} s_{b-k} + s_{b-k} s_b.$$

The value of u is maximized by setting the free spins $s_{a+k} = s_a$ and $s_{b-k} = s_b$. If $s_a = s_b$, the center term is 1 and $u_{\max} = 3$. For $s_a \neq s_b$, the center term is -1 and $u_{\max} = 1$. The value of u is minimized by setting $s_{a+k} = -s_a$ and $s_{b-k} = -s_b$. If $s_a = s_b$, the center term is 1 and $u_{\min} = -1$. If $s_a \neq s_b$, the center term is -1 and $u_{\min} = -3$. Now let us assume that (25) holds for some odd $n \geqslant 3$ and consider a free chain

$$u = s_a s_{a+k} + s_{a+k} s_{a+2k} + \cdots + s_{b-2k} s_{b-k} + s_{b-k} s_b$$

with $n + 2$ terms. To maximize u, we set $s_{a+k} = s_a$ and $s_{b-k} = s_b$, and the remaining free chain has n terms. Applying (25), we get $u_{\max} = n + 2$ if $s_a = s_b$ and $u_{\max} = n$ if $s_a \neq s_b$. The induction step for $u_{\min}$ is obvious.

Since the proof for even n is very similar, it is omitted here. We only mention that the base case ($n = 2$) corresponds to the 'cancellation' and 'reinforcement' used by Prestwich to improve the bound (12).

Now (24) and (25) tell us how to compute $u_{\min}$ and $u_{\max}$ for each individual free chain. The corresponding values $U_k^{\min}$ and $U_k^{\max}$ are obtained by summing over all free chains that contribute to u_k.

Every branch of the search tree that can be pruned according to the combined bound (17) (or any other relaxation of (9)) is also pruned by the tight bound (21), but the tight bound allows us to prune additional branches. Hence the number of recursive calls with the tight bound can not be larger than the number of calls with any other bound based on (9). What we observe is that for $N \leqslant 66$ the number of calls for the tight bound is in fact strictly smaller than that for the combined bound. A numerical fit to the existing data yields a scaling of $\Theta(1.727^N)$ for the tight bound, compared to $\Theta(1.729^N)$ for the combined bound, see figure 2. This difference is too small to tell whether the tight bound actually provides an exponential speedup or not. In fact, if one looks at the ratio of the number of calls for the combined bound

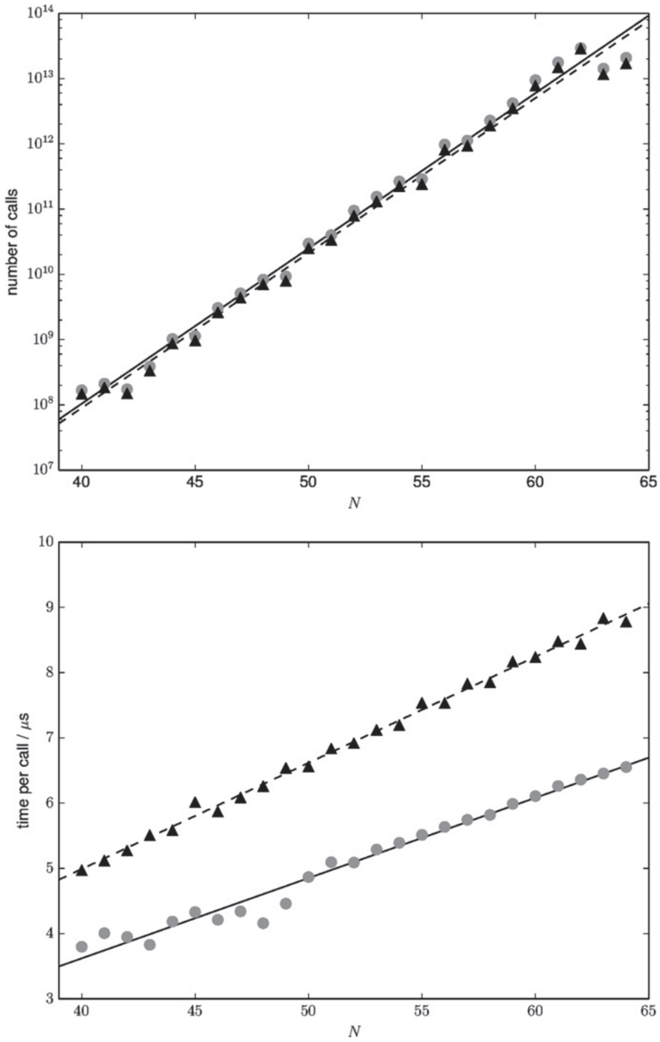

Figure 2. Branch and bound algorithm with the combined bound (17) (circles) and with the tight bound (21) (triangles). The number of recursive calls (top) scales like $\Theta(b^N)$. A numerical fit to the existing data yields $b = 1.729$ (solid line) for the combined bound and $b = 1.727$ (dashed line) for the tight bound, but this small difference is caused by a non-exponential reduction of the number of calls, see figure 3. The CPU time per call (bottom) is linear in N for both bounds.

divided by the number of calls for the tight bound, one observes that the speedup factor grows linearly with N, not exponentially (figure 3). Since the time per call scales linearly for both bounds (figure 2 bottom), a reduction of the number of calls that grows with N implies that the tight bound will asymptotically outperform the combined bound.

For the values of N considered in this paper, however, the absolute computational costs per call matter. And here the simpler combined bound (17) is faster, see figure 2 (bottom). If we extrapolate the number of calls and the time per call to $N = 66$, we get a running time of roughly 12 600 CPU days for the combined bound but 14 300 CPU days for the tight bound. This is why we used the weaker combined bound for all the new solutions (exact and

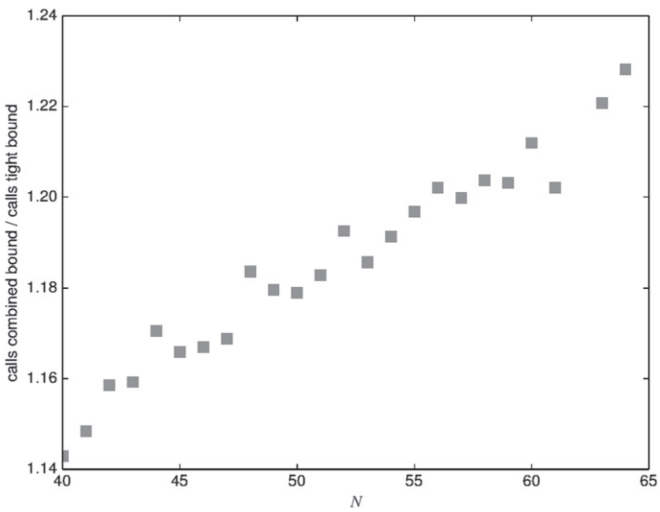

Figure 3. Number of calls for the combined bound (17) divided by the number of calls for the tight bound (21). For the values of N considered here, the speedup due to the tight bound seems to grow linearly with N.

skewsymmetric) reported in this paper. Note that the time per call depends considerably on the implementation. It might well be possible to implement the tight bound such that it outperforms the combined bound already for the values of N considered here. In any case, the measured running times illustrate that we need to parallelize the computation if we do not want to wait 35 years for the $N = 66$ LABS solution.

3.2. Symmetry and parallelization

The correlations C_k (1) are unchanged when the sequence is complemented or reversed. When alternate elements of the sequence are complemented, the even-indexed correlations are not affected, the odd-indexed correlations only change sign. Hence, with the exception of a small number of symmetric sequences, the 2^N sequences will come in classes of eight which are equivalent. The total number of non-equivalent sequences is slightly larger than 2^{N-3}.

The m left- and m rightmost elements of the sequence can be used to parameterize the symmetry classes. The total number $c(m)$ of symmetry classes that can be distinguished by m left- and m right-border elements reads

$$c(m) = 2^{2m-3} + 2^{m-2+(N \bmod 2)}. \tag{26}$$

We derive this formula in the appendix, where we also describe how to compute the values of the $2m$ boundary spins that represent each symmetry class.

The symmetry classes can be enumerated independently, which allows us to parallelize the computation. For our largest system ($N = 66$) we used $c(m = 10) = 131\,328$ symmetry classes that we searched in parallel on various computers with number of computing cores ranging from 8 to 5700. In principle, the branch and bound algorithm requires some communication between the parallel tasks since every task should know the lowest energy found so far by other tasks to compare it to the bound. We avoid this communication completely by using a static value for this reference energy: the lowest energy found by heuristic searches. In all cases we considered, this value turned out to be the true minimum energy.

4. Results and conclusions

We have used the branch and bound algorithm with the combined bound to compute all sequences with maximum merit factor for $N \leqslant 66$, see tables 1 and 2. The previous record was $N \leqslant 64$, obtained with the Wiggenbrock bound (16) and using 18 GPUs [30]. For the perfomance measurements for $40 \leqslant N \leqslant 64$ shown in figure 2 we have used a Linux cluster with a collection of Intel® Xeon® CPUs: 10× E5-2630 (at 2.30 Ghz), 10× E5-2630 v2 (at 2.60 Ghz) and 2× E5-1620 (at 3.60 Ghz) with a total of 248 (virtual) cores. On this machine, the computation for $N = 64$ took about a week (wallclock time). As one can see in figure 2, the solution of $N = 63$ and $N = 64$ involves a surprisingly low number of calls and took therefore less time than actually expected.

Note that with our algorithm systems of size $N \leqslant 43$ can be solved in less than an hour on a laptop.

For $N = 65$ and $N = 66$ we used a variety of computing machinery that makes an accurate determination of 'single CPU time' impossible. For $N = 65$ and 66, the equivalent wallclock time on our benchmark cluster is roughly 32 and 55 d.

Tables 1 and 2 show all sequences (except those related by symmetries) with maximum merit factors up to $N = 66$ in run-length encoding, i.e. the digits specify the length of runs of equal spins. We use $a = 10$, $b = 11$ etc for runs of spins that are longer than 9.

We have used our branch and bound algorithm also to find all skewsymmetric sequences with maximum merit factor up to $N = 119$. The previous record was $N \leqslant 89$ [29]. Table 3 shows the skewsymmetric sequences with maximum merit factor as far as they are not listed in tables 1 and 2. Skewsymmetric merit factors marked with $\star$ are known to be not maximal. We know this either from exhaustive enumerations (for $N \leqslant 65$) or from heuristic searches that have yielded non skewsymmetric sequences with larger merit factors.

Figure 4 shows the ratio of the maximum merit factors of skewsymmetric and general sequences for $N \leqslant 119$. In 20 out of 58 cases the skewsymmetric subset does not contain a maximum merit factor sequence. Note that the values of F_N for $N > 66$ are from heuristic searches, but we believe that these values are the true maximum merit factors. But strictly speaking, the gray symbols in figure 4 are only upper bounds for the ratio F_N^{skew}/F_N. The available data seems to indicate that roughly two thirds of all odd values of N have skew-symmetric maximum merit factor sequences. Figure 4 also suggests that

$$\liminf_{N\to\infty} \frac{F_N^{\text{skew}}}{F_N} = 1. \tag{27}$$

We think that the branch and bound approach based on the relaxation (9) can be used to solve the LABS problem for $N > 66$ by devoting more compute cores and more CPU time. Improving the implementation to reduce the constant factor in the $\Theta(N\, b^N)$ scaling can also help. Solving systems significantly larger than $N = 66$, however, requires a stronger bound than (9), i.e., a bound that takes into account the fact that the C_k are not independent. Or a completely new approach other than branch and bound.

Appendix. Symmetry

To find the exact number of symmetry classes in the LABS problem we need some group theory. The operators R (reverse), C (complement) and A (alternate complement) act on the sequences, leaving the energy invariant. Together with the identity operator I these operators generate a group G of order 8. The structure of G depends on N being odd or even.

Table 1. All optimal low autocorrelation binary sequences for $N \leqslant 47$ modulo symmetries.

N	E	F_N	Sequences	Skew
3	1	4.500	21	×
4	2	4.000	112	
5	2	6.250	311	×
6	7	2.571	141	
			123	
			312	
			1113	
7	3	8.167	1123	×
8	8	4.000	32111	
			31121	
9	12	3.375	311121	
			42111	×
			32211	×
			31122	
10	13	3.846	42211	
			52111	
			311122	
			41122	
			33121	
11	5	12.100	112133	×
12	10	7.200	4221111	
			4111221	
13	6	14.083	5221111	×
14	19	5.158	41112221	
			6221111	
			5222111	
			33111212	
			41111222	
			42211112	
			5221112	
			5311121	
15	15	7.500	52221111	×
			33131211	×
16	24	5.333	225111121	
			6322111	
			313311211	
			2131441	
17	32	4.516	252211121	×
			44121311	
			4221211112	
			36111221	
			2122411112	
			2112113132	
18	25	6.480	441112221	
			511211322	
19	29	6.224	4111142212	
20	26	7.692	5113112321	

Table 1. (Continued.)

N	E	F_N	Sequences	Skew
21	26	8.481	27221111121	×
22	39	6.205	51221111233	
			632111112211	
			511111212232	
23	47	5.628	212121111632	
			83211112211	
			314121131132	
24	36	8.000	2236111112121	
25	36	8.681	337111121221	
26	45	7.511	21212111116322	
			63231111121211	
			32361111121211	
27	37	9.851	34313131211211	×
28	50	7.840	34313131211212	
29	62	6.782	212112131313431	×
			323711111212211	×
30	59	7.627	551212111113231	
			461212111113231	
31	67	7.172	7332212211112111	
32	64	8.000	71112111133221221	
33	64	8.508	742112111111122221	
34	65	8.892	842112111111122221	
35	73	8.390	7122122111121111332	
36	82	7.902	3632311131212111211	
37	86	7.959	844211211111122221	
38	87	8.299	8442112111111122221	
39	99	7.682	82121121234321111111	×
			23241171111141122121	×
40	108	7.407	44412112131121313131	
41	108	7.782	343111111222281211211	×
42	101	8.733	313131341343112112112	
43	109	8.482	1132432111117212112213	×
44	122	7.934	525313113111222111211121	
45	118	8.581	82121121231234321111111	×
46	131	8.076	823431231211212211111111	
			821211212312343211111111	
			73235111112132122112121	
47	135	8.181	923431231211212211111111	×
			429422222112111111122111	×
			411121114131131312421242	
			383422132211212111111211	×
			236331611113121211112121	×
			212a211212342111111111231	×

For N even, $G = G_e$ is non-abelian and isomorphic to the dihedral group D_4, the symmetry group of a square plate, generated by a 90° rotation and a flip. The elements of G_e are $\{I, R, C, A, RA, AR, RC, AC\}$. The group elements act on a sequence s. Let

Table 2. All optimal low autocorrelation binary sequences for $48 \leqslant N \leqslant 66$ modulo symmetries.

N	E	F_N	Sequences	Skew
48	140	8.229	3111111832143212221121121	
49	136	8.827	2151313112241122411141141	
			3337313221312111112121211	×
50	153	8.170	2151313112241122411141142	
			72542221311111132111211211	
			4337313221312111112121211	
51	153	8.500	23432111141313116212112121	×
52	166	8.145	51161212121111131223123332	
53	170	8.262	451131113325131222111211121	
			22b44222211211211111111221	×
54	175	8.331	35622514121211222211111121	
55	171	8.845	921212321211432123321111111	×
			3232a41124112111111112212211	×
56	192	8.167	761223112324111113211212211 1	
57	188	8.641	3323263111112712111122122121 1	×
58	197	8.538	1111131232138142121132432112	
59	205	8.490	77241224211223112211111211111 1	×
			613212312111111311234122112124 2	
60	218	8.257	7611121411111311242113222112 22	
			2222221113111142441611211611 21	
61	226	8.232	3141623312111111311121256212 11	
62	235	8.179	3232321111171115411215112221 22	
			a2322321213531122111111211311 2	
63	207	9.587	21221221271111151112114311142 2321	
64	208	9.846	21221221271111151112114311142 2322	
65	240	8.802	32322411134112111511111721221 2212	
66	257	8.475	2112111211222b22211111111122 24542	

$$G(s) := \{g(s) : g \in G\} \tag{A.1}$$

denote the orbit of s, i.e. the set of all spin sequences that can be generated from s by application of the group elements. The length of the orbit is $|G(s)|$. The orbits partition the set of all spin sequences in the symmetry classes we want to count. If all orbits were of length 8, we would have 2^{N-3} symmetry classes. Unfortunately there are orbits of smaller length, to wit

$$G(++++) = \{++++, ----, -+-+, +-+-\}.$$

The true number c of orbits is given by Burnside's Lemma

$$c = \frac{1}{|G|}\sum_{g \in G} |\mathrm{Fix}(g)|, \tag{A.2}$$

where $\mathrm{Fix}(g) = \{s : g(s) = s\}$ denotes the set of all sequences s that are fixed points of g. The group elements A, C, AR, RA and AC can not fix a sequence, but R and RC can. Sequences with $s_j = s_{N+1-j}$ are fixed by R, sequences with $s_j = -s_{N+1-j}$ are fixed by RC, and there are $2^{N/2}$ sequences of each type. Hence

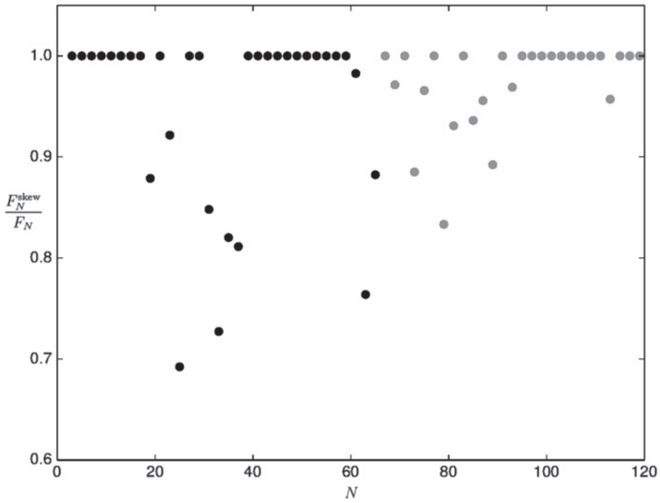

Figure 4. Ratio of maximum merit factors: skewsymmetric F_N^{skew} versus general F_N. Black symbols are exact, gray symbols are based on lower bounds for F_N, which are believed to be exact.

$$c_{N\text{even}} = \frac{1}{8}(|\text{Fix}(I)| + |\text{Fix}(R)| + |\text{Fix}(RC)|) = 2^{N-3} + 2^{N/2-2}. \tag{A.3}$$

For N odd, the group $G = G_o$ is again of order 8, but this time it is abelian. Group elements are $G_o = \{I, R, C, A, RC, RA, CA, RCA\}$, and $g^2 = I$ for all $g \in G_o$. G_o is isomorphic to the reflection-symmetry group of the cube. If $(N-1)/2$ is odd, only R, RA and I have fixed points. Sequences that are fixed by R have $s_j = s_{N+1-j}$ with arbitrary center spin $s_{(N+1)/2}$. There are $2 \cdot 2^{(N-1)/2}$ such sequences. The same number of sequences are fixed by RA. Hence

$$c_{N\text{odd}} = \frac{1}{8}(|\text{Fix}(I)| + |\text{Fix}(R)| + |\text{Fix}(RA)|) = 2^{N-3} + 2^{(N-1)/2-1}. \tag{A.4}$$

If $(N-1)/2$ is even, only I, R and RCA have fixed points, and their numbers are the same as in (A.4). Combining (A.3) and (A.4) provides us with

$$c_N = 2^{N-3} + 2^{(N-1)/2-2+(N \bmod 2)}. \tag{A.5}$$

This is the total number of symmetry classes if we consider all elements of the sequence. If we only consider the m leftmost and m rightmost elements, the arguments are similar. For N even, the symmetry group $G_e = \{I, R, C, A, RA, AR, RC, AC\}$ acts only on the $2m$ elements, and only I, R and RC have fixed points. Hence

$$c(m) = 2^{2m-3} + 2^{m-2} \qquad N \text{ even}. \tag{A.6}$$

For N odd, the symmetry group is again $G_o = \{I, R, C, A, RC, RA, CA, RCA\}$, but this time I, R, RC, RA and RCA have fixed points:

$$c(m) = 2^{2m-3} + 2^{m-1} \qquad N \text{ odd}. \tag{A.7}$$

Combining (A.6) and (A.7) provides us with (26).

The $c(m)$ symmetry classes can be uniquely parameterized by the values of the $2m$ boundary spins. Consider the list of all 2^{2m} possible configurations of the boundary spins. For each such configuration compute $G(s) = G_e(s)$ (for N even) or $G(s) = G_o(s)$ (for N odd). If s

Table 3. All optimal skewsymmetric low autocorrelation binary sequences for $N \leqslant 119$ as far as they are not listed in table 1 or 2. Merit factors marked with $\star$ are known to be not maximal, either from exhaustive enumeration (for $N \leqslant 65$) or from heuristic searches (for $N \geqslant 67$).

N	E	F_N^{skew}	Sequences
19	33	5.470⋆	2113114141, 3513111211
23	51	5.186⋆	272221111121, 336111121211, 343131211211, 732212111111
25	52	6.010⋆	6332121211111
31	79	6.082⋆	6212211423211111
33	88	6.188⋆	84212321121111111, 22742211211111221
35	89	6.882⋆	472322122111112111, 552212232211121111
37	106	6.458⋆	2492222111111121121
61	230	8.089⋆	2121111221121411111122811342631
63	271	7.323⋆	a11211121122212223224541111111111
			21242131111311112112461613211231
			23111131111323531211121221616121
65	272	7.767⋆	414411126121313133111125112113111
			231134321111114222211821211214121
			221111121211111132311224122183721
			2112111211222b2221111111112224541
67	241	9.313	b412323441121121221231121111111111
			6216121225331212111223311113211111
			2454222111111111222b22211211121121
69	282	8.441⋆	211111111211321221211211443232214b1
71	275	9.165	241244124172222111113112311211231121
73	348	7.657⋆	2111211211111221131113213132151427451
			22c74422222211211211111211111111111221
75	341	8.248⋆	23231233481611113111111211212123122121
77	358	8.281	512174112122112221322423411211111331111
79	407	7.667⋆	4361113231311213321213413122151111212111
			3121312121411112131112112451361133313311
			2131211221311121211121131141453513243131
			2129214121111212131124133531132111111231
			2111213111121123314261111221131212461351
81	400	8.201⋆	53611132313112133212134131221511112121111
83	377	9.137	323633231172611112211111412212121111212211
85	442	8.173⋆	3912523121213351112121333122111231111111211
87	451	8.391⋆	43114242215111132131313216111322112211412111
89	484	8.183⋆	231143113311111143233221212212118121412114121
			231433161111121421112123521137111131212113121
91	477	8.680	2121416112211211111211321222321474241111311331
93	502	8.615⋆	91252112312341122322122411212312421112311111111
			21121213121261171226221111223111114111123313341
			25523581113122413112231511111121112122111211121
95	479	9.421	322322358115111351112151114111111211121222122211
97	536	8.777	5111415321132221132143121132142221421211131151111
99	577	8.493	5255212212a31122411224121111111123232111211122111 1
101	578	8.824	6255212212a311224112241211111111232321112111221111 1
103	555	9.558	245268122221311122511122513222311111121111221112112 1
105	620	8.891	a121112112141121311213222122322213413411345311111111 1
107	677	8.456	22731183111122411334222112121411226121111114121111122 1
109	662	8.974	334111111243114111113322225111222221217114121128112121 1

Table 3. (Continued.)

N	E	F_N^{skew}	Sequences
111	687	8.967	21111323331321111135114211332121421141112172131212122161
113	752	8.490*	4555122142121212c12223111111111123332113231112111211121112111
			231332171323311541212112134331121114121221311111321213121
115	745	8.876	551113514531112211312122222223314251211121131112151112111 1
117	786	8.708	3711731211122111113322242411221122221217253121111141111211
119	835	8.480	3121614121221234111211113141113215113165112123233113111133 11

is not the lexicographically smallest element in $G(s)$, remove it from the list. The remaining elements are a unique representation of the symmetry classes.

References

[1] Golay M J E 1972 A class of finite binary sequences with alternate autocorrelation values equal to zero *IEEE Trans. Inf. Theory* **IT-18** 449–50

[2] Beenker G F M, Claasen T A C M and Hermens P W C 1985 Binary sequences with a maximally flat amplitude spectrum *Philips J. Res.* **40** 289–304

[3] Pasha I A, Moharir P S and Rao N S 2000 Bi-alphabetic pulse compression radar signal design *Sãdhanã* **25** 481–8

[4] Shapiro I I, Pettengill G H, Ash M E, Stone M L, Smith W B, Ingalls R P and Brockelman R A 1968 Fourth test of general relativity *Phys. Rev. Lett.* **20** 1265–9

[5] Littlewood J E 1968 *Some Problems in Real and Complex Analysis* (Lexington, MA: D C Heath & Co)

[6] Borwein P 2002 *Computational Excursions in Analysis and Number Theory* (New York: Springer)

[7] Bernasconi J 1987 Low autocorrelation binary sequences: statistical mechanics and configuration space analysis *J. Phys.* **48** 559–67

[8] Krauth W and Mézard M 1995 Aging without disorder on long time scales *Z. Phys.* B **97** 127–31

[9] Bouchaud J P and Mézard M 1994 Self induced quenched disorder: a model for the glass transition *J. Phys.* I **4** 1109–14

[10] Marinari E, Parisi G and Ritort F 1994 Replica field theory for determistic models: I. Binary sequences with low autocorrelation *J. Phys. A: Math. Gen.* **27** 7615–45

[11] Jedwab J 2005 A survey of the merit factor problem for binary sequences *Sequences and Their Applications Proc. SETA 2004* (*Lecture Notes in Computer Science* vol 3486) ed T Helleseth *et al* (Berlin: Springer) pp 30–55

[12] Høholdt T 2006 The merit factor problem for binary sequences *Applied Algebra, Algebraic Algorithms and Error-Correcting Codes* (*Lecture Notes in Computer Science* vol 3857) ed M P C Fossorier *et al* (Berlin: Springer) pp 51–9

[13] Barker R H 1953 Group synchronizing of binary digital systems *Communication Theory* ed J Willis (London: Butterworths) pp 273–87

[14] Turyn R and Storer J 1961 On binary sequences *Proc. Am. Math. Soc.* **12** 394–9

[15] Schmidt K-U and Willms J 2015 Barker sequences of odd length 2015 *Designs, Codes and Cryptography* (doi:10.1007/s10623-015-0104-4)

[16] Leung K H and Schmidt B 2012 New restrictions on possible orders of circulant Hadamard matrices *Des. Codes Cryptogr.* **64** 143–51

[17] Golay M J E 1982 The merit factor of long low autocorrelation binary sequences *IEEE Trans. Inf. Theory* **IT-28** 543

[18] Newmann D J and Byrnes J S 1990 The l^4 norm of a polynomial with coefficients ±1 *Am. Math. Mon.* **97** 42–5

[19] Jedwab J, Katz D J and Schmidt K-U 2013 Advances in the merit factor problem for binary sequences *J. Comb. Theory* A **120** 882–906

[20] Jedwab J, Katz D J and Schmidt K-U 2013 Littlewood polynomials with small l^4 norm *Adv. Math.* **241** 127–36

[21] Bošković B, Brglez F and Brest J 2014 Low-autocorrelation binary sequences: on the performance of memetic-tabu and self-avoiding walk solvers arXiv:1406.5301
[22] Savage C 1997 A survey of combinatorial Gray codes *SIAM Rev.* **39** 605–29
[23] Groot C d, Würtz D and Hoffmann K H 1992 Low autocorrelation binary sequences: exact enumeration and optimization by evolutionary strategies *Optimization* **23** 369–84
[24] CSPLib: a problem library for constraints (www.csplib.org)
[25] Moore C and Mertens S 2011 *The Nature of Computation* (Oxford: Oxford University Press) www.nature-of-computation.org
[26] Mertens S 1996 Exhaustive search for low-autocorrelation binary sequences *J. Phys. A: Math. Gen.* A **29** L473–81
[27] Bauke H and Mertens S 2004 Ground states of the Bernasconi model with open boundary conditions (http://ovgu.de/mertens/research/labs/open.dat)
[28] Prestwich S D 2007 Exploiting relaxation in local search for LABS *Ann. Oper. Res.* **156** 129–41
[29] Prestwich S D 2013 Improved branch-and-bound for low autocorrelation binary sequences arXiv:1305.6187
[30] Wiggenbrock J 2010 Parallele Optimierungsstrategien des LABS-problems in einem GPU-grid *Bachelor's Thesis* Fachhochschule Südwestfalen