Received: 21 October 2019 | Revised: 4 October 2020 | Accepted: 14 December 2020
DOI: 10.1002/dac.4719

RESEARCH ARTICLE

WILEY

Efficient method for constructing optimized long binary spreading sequences

Sabrina Boukerma[1] | **Khaled Rouabah**[2] | **SalahEddine Mezaache**[2] | **Salim Atia**[2]

[1]Electronics Department, University of Mohamed BOUDIAF-M'Sila, M'Sila, Algeria

[2]LETA, Electronics Department, University of Mohamed El-Bachir El-Ibrahimi–Bordj Bou Arréridj, El Anceur, Algeria

Correspondence
Sabrina Boukerma, Electronics Department, University of Mohamed BOUDIAF-M'Sila, M'Sila, Algeria.
Email: sabrina.boukerma@univ-msila.dz

Summary

In this paper, we propose an efficient method for generating two types of novel optimized long binary spreading sequences (OLBSS) with improved autocorrelation function (ACF) properties. The first type is constructed from concatenated short binary subsequences belonging to the same code family, such as Walsh Hadamard and Gold subsequences, provided that their cross-correlation functions (CCFs) have good properties. The second category uses the same subsequences but which are rather interlaced. Here, the number and size of the subsequences are related to the chosen length of the final constructed long sequence and the desired performances. The realization of the OLBSSs is achieved using two different optimization techniques, namely, the genetic algorithms (GAs) and particle swarm optimization (PSO) method. The simulation results, based on MATLAB tool, have shown that the proposed long sequences, composed of Walsh–Hadamard subsequences and optimized by the GA, have better ACF properties compared to the original Gold, Weil, and random sequences of the same length.

KEYWORDS

autocorrelation function, genetic algorithms, Gold, long binary spreading-sequences, particle swarm optimization, Walsh–Hadamard

1 | INTRODUCTION

In telecommunications applications, many multiple access schemes exist to allow multiple users to share a single communication medium.[1] The most basic of them is the code division multiple access (CDMA) that is based on direct sequence spread spectrum (DSSS) communication technique. The basic idea of CDMA is that the data sequence to be transmitted is multiplied by a pseudo noise (PN) sequence, known only by users. The direct consequence of this coding operation is the spreading of the informative signal's power spectral density (PSD) on the whole band of the PSD of the PN code that is much wider than the minimum required for the transmission of the informative signal.[2,3] At the receiving end, the received signal is correlated with a locally generated replica of the transmitted code; the original information is then recovered when the two correlated signals are synchronized, eliminating at the same time the influence of the PN code.

The commonly used spreading sequences in CDMA are m-sequences,[4,5] Gold sequences,[6] small and large Kasami sequences,[7] Weil sequences,[8] random codes or memory sequences,[9] chaotic sequences,[10] and zero correlation zone (ZCZ) sequences[11,12] used to eliminate multipath and co-channel interference in fading channel.[11] The m-sequences, Gold sequences, and small and large Kasami sequences are well-known as families of binary sequences with good CCF properties.[5] They are all generated from linear feedback shift register (LFSR).[13] The Gold sequences, used especially in some global positioning system (GPS) applications, are the best-known family of m-sequences.[14] The Weil codes, constructed for any prime code length, are another type of sequences that are also used in some GPS applications, especially in the L1C band.[15] The random codes, known similarly as memory codes, are used by Galileo navigation system for the E1 open service (OS) and E6 bands.[15]

A family of spreading sequences whose mutual CCFs are zero for any time shift is known as a family of orthogonal sequences.[16] We can give as examples, Walsh–Hadamard sequences,[17] orthogonal Gold sequences, and orthogonal variable spreading factor (OVSF) sequences.[16,18] Orthogonal sequences have been firstly introduced for 3G communication systems, such as UMTS-UTRA,[19] W-CDMA,[20] and TD-SCDMA.[21] Walsh–Hadamard sequences have many popular applications in current wireless communications standards such as IS-95,[22] CDMA2000,[23] and WCDMA.[24] Other less widely cited spreading codes include Kronecker sequences,[25] Barker sequences,[26,27] GMW sequences,[28] Bent sequences,[29] No sequences,[30] and wavelet sequences.[31] All of these well-known sequences have strongly contributed to the modernization and the expansion of the global navigation satellite system (GNSS) fields of applications.[32–34] Nowadays, the emergence of several GNSS systems, such as the American GPS, the Russian GLONASS (GLObal'naya NAvigatsionnaya Sputnikovaya Sistema), the Chinese COMPASS, and the new European Galileo, has greatly contributed to the design of this type of codes.[35] In fact, each of these systems contains a spatial segment consisting of a constellation of satellites each emitting a set of PN codes.[36] These codes are the fundamental elements in all GNSS systems and represent the tool that permits to a GNSS receiver to distinguish one satellite from another.[37] For example, original GPS signals are composed of the two carrier frequencies L1 (1575.42 MHz) and L2 (1227.6 MHz), each modulated by data navigation message and one digital code that consists of Coarse Acquisition (C/A) code or Precise (P) Code.[38,39,14] C/A code, which is a Gold code, is used for civilian applications while P code, belonging to the same category of codes, is reserved for military use and authorized civilian users. The current GPS codes, used for the L1C civil signal, are quite different from the original ones[40] since they fit to the class of Weil codes that are based on Legendre sequences.[41,42] Galileo spreading codes are constructed by tertiary codes, consisting of a modulation of two codes: a primary code and a secondary code.[43] Galileo will broadcast, for the first time, the so-called random codes, which are codes optimized in a highly multidimensional space to make them look as random as possible.[15,44] The BeiDou Navigation Satellite System (BDS), also named COMPASS Navigation Satellite System (CNSS), uses Gold codes in all its frequency bands, specifically E2, E5b, and E6.[45] The E2 and E5b codes are identical and are based on a truncated 11th order Gold code of 2046 bits. The E6 has a length of 10 230 bits and consists of a concatenation of two Gold code segments. Both segments are based on truncated 13th order Gold codes having the same code polynomials but with different initial states. The E2, E5b, and E6 primary codes are modulated with a 20-bit Neuman–Hoffman code that is used as secondary code.[46] The ranging codes of the GLONASS system are maximum length sequences (m-sequences) generated by using maximal LFSR. They are periodical sequences with a period of 1 ms and a bit rate of 511 kilobits per second.[47]

The required characteristics of all these codes consist, in the first place, in the obtainability of a great number of codes belonging to the same family. Secondly, each of these codes must have an impulsive ACF and a random shape in order to minimize multipath effects and ensure robustness against noise.[48] Thirdly, every two codes of the same family must have zero CCF values (close to 0 at all-time shifts) in order to minimize the multiple access interference (MAI) between different users.[49] Fourthly, they must support variable data rates.[49] Finally, the obtained codes must be long and must have all the previous characteristics. This last feature is the most important one, since a longer length, for DSSS communications, provides better separation of signals from different sources and increases thus robustness against interference and noise. Unfortunately, it is very difficult to achieve the generation of such codes having all these characteristics.

In recent years, different approaches, for code generation and optimization, have been proposed in scientific literature.

For example, Ren[50] proposed a method for the construction of long period sequences from the addition of m-sequences with pairwise-prime linear spans (AMPLS). The AMPLS sequences have excellent ACF properties, balanced statistics, and large complexity. However, the major disadvantage of the AMPLS sequences is the presence of limited number of m-sequences.

Another method, given in Sarayloo et al.,[51] is employed to generate long binary sequences based on the use of a number of shorter De Bruijn sequences (DBSs).[52] DBSs sequences are a kind of important pseudorandom sequences with many desirable properties such as long period, large linear complexity, and the best pattern distribution.[53] Different constructions of DBSs were studied in previous studies,[54–56] and their linear complexity has been thoroughly investigated in other works.[57,58] However, this method is also limited by the small number of shorter De Bruijn subsequences that can be generated.

In 2017, Meng and Yan found two constructions of binary interleaved sequences of period 4N by selecting appropriate shift sequences, subsequences, and complement sequences.[59] These sequences have low ACF under certain conditions with less linear complexity which remains to be solved.

Zhao et al.[60] proposed a unified framework to design low ACF sequences. They optimize a unified metric over a general constraint set via using the majorization–minimization (MM) method, which is iterative and enjoys guaranteed convergence to a stationary solution. However, this method is limited by the grid size needed by the framework, which is a sensitive user parameter.

Other studies[61–63] present best currently results of the global search for binary sequences that are optimal over the minimum peak side lobe (PSL) criterion. In Dimitrov et al.,[63] a simple and efficient algorithm, based on heuristic search by shotgun hill climbing, is used to construct binary sequences with lengths between 106 and 300. However, such sequences' lengths remain too short for use in DSSS communications.

Alaee-Kerahroodi et al.[64] provide another way to design a set of binary sequences with good aperiodic/periodic auto- and cross-correlation functions for multiple-input multiple-output (MIMO) radar systems. The proposed method is based on a minimization of weighted sum of PSL and integrated side lobe (ISL) level with the binary element constraint at the design stage and uses the block coordinate descent (BCD) framework to minimize the multidimensional objective function

In their study,[65] Bose et al. proposed an efficient computational framework for designing sequences with both good ACF and good distribution properties. This latter method uses the cyclic algorithm-new (CAN) that is a computational framework[66] based on the fast Fourier transform (FFT) operations and can generate up to $N{\sim}10^6$ very long sequences or even longer in small time frames. Furthermore, Bose and Soltanalian proposed a construction method of binary sequences with asymptotically optimal PSL growth from the sequences set with good correlation properties such as Gold, Kasami, Legendre, and Weil.[67] The proposed method constructs the binary sequences through a nonconvex quadratic program that can be handled in polynomial time.

Soltanalian et al.[68] proposed an extension of the CAN algorithm,[66] named CANARY, for designing complementary sets of sequences. The CANARY algorithm, which works in the frequency domain, is used to design aperiodic binary sequences with good ACF properties. Furthermore, 1bCAN and 1bPeCAN proposed in Lin et al.[69] that combine the notion of converging functions with the CAN and periodic CAN (PeCAN)[70] frameworks can be used to design very long binary sequences (up to $N{\sim}10^6$or even longer on an ordinary laptop) with good ACF properties. 1bCAN shows a superior performance in terms of merit factor (MF) for aperiodic binary sequence designs compared with CANARY algorithms. However, all these sequences are not valid for applications based on periodic ACF.

In this paper, two types of new OLBSSs with better periodic ACF properties are proposed. They are both obtained by constructing long spreading sequences from a number of shorter binary subsequences of same length and belonging to the same family such as Walsh–Hadamard sequences or Gold sequences. In the first type, the subsequences are concatenated, while in the second type, they are interlaced. Two different algorithms have been used, including GA[71,72] and PSO method,[73,74] to select the optimal position of each subsequence in the first type and that of the first bit of each subsequence in the second type.

This paper is organized as follows: Section 2 describes the generation principle and the basic optimization concepts of the proposed sequences using GA and PSO. The choice of the fitness function, the mathematical proof, and the experimental results are presented and discussed in Section 3. Finally, we end up by a conclusion.

2 | GENERATION PRINCIPLE OF THE OPTIMIZED LONG SEQUENCES

In what follows is given a brief presentation of the generation principle and the characteristics of both aforementioned types of proposed spreading long sequences, resulting from the concatenation/interlacement of shorter binary subsequences. Subsequently, we present in detail the principle of generating the OLBSS.

2.1 | Walsh–Hadamard and Gold sequences

The name of Walsh–Hadamard sequences comes from the American mathematician Joseph Leonard Walsh and the French mathematician Jacques Hadamard. Walsh codes can be generated from Hadamard matrix of any order that is of power of two (2^N). The rows (or columns) of the resulted matrix of order 2^N constitute the Walsh sequences. The Hadamard matrix of desired length can be generated by the following recursive procedure:

$$H_N = \begin{bmatrix} H_{N/2} & H_{N/2} \\ H_{N/2} & \bar{H}_{N/2} \end{bmatrix} \tag{1}$$

With
$H_1 = 1$, H_2 can be given from 1 as follows:

$$H_2 = \begin{bmatrix} H_1 & H_1 \\ H_1 & \bar{H}_1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{2}$$

The ACF (which is a measure of the similarity between a sequence $c[t]$ and its time shifted replica $c[t - \tau]$) of a sequence $c(t)$ is defined as follows:

$$R_{cc}(\tau) = \int_{-\infty}^{+\infty} c(t)c(t-\tau)dt \tag{3}$$

Figure 1 shows an example of the ACF of a Walsh–Hadamard sequence of length 64 bits. It can be seen that the ACF presents several peaks.

Gold sequences have been proposed by Gold in 1967 and 1968.[15,19,20] The latter ones are constructed by the XOR of two m-sequences u and v of the same length (N). The family of this type of codes is defined as follows:

$$G(u,v) = \left\{u, v, u \oplus v, u \oplus T \cdot v, u \oplus T^2 \cdot v, \ldots, u \oplus T^{N-1} \cdot v\right\} \tag{4}$$

T^k denotes the operator which shifts vectors cyclically to the left by k places, u and v are m-sequences of the same length, and $\bigoplus$ is the exclusive OR operator.

The resulting sequence $G(u,v)$ is not of maximum length but always of the same length (N). Therefore, it is possible to generate a total of N+2 sequences of length N.

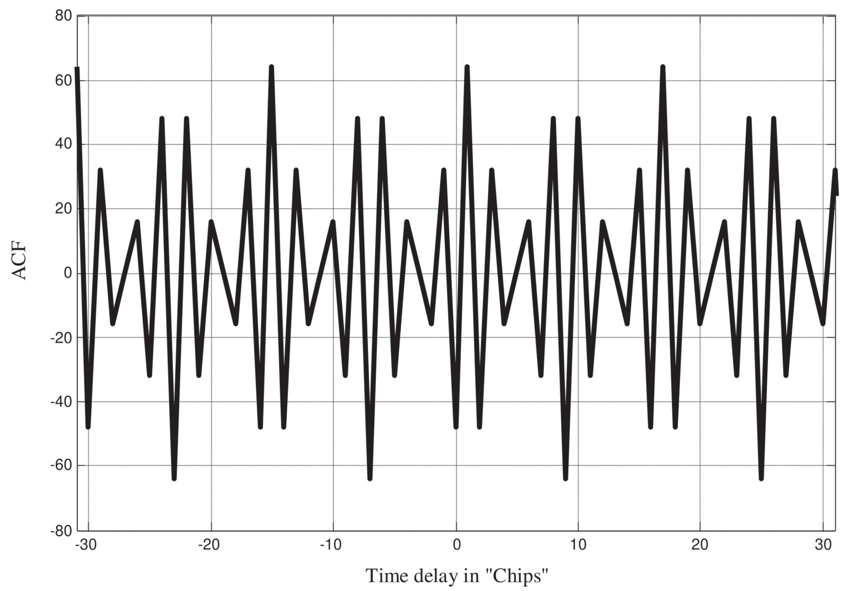


FIGURE 1 ACF of a Walsh–Hadamard sequence of length 64 bits

An example of the ACF of a Gold sequence of length 1023 bits is plotted in Figure 2.

From Figures 1 and 2, we note that Gold sequence has better ACF properties compared to those of Walsh–Hadamard.

2.2 | Proposed long binary sequence

According to the way the subsequences are used (concatenation or interlacement), we can construct two types of long binary sequences S from M different shorter subsequences $A_i(i = 1, \ldots M)$ of same length l_s that belong to the same family.

1. *First type (concatenation)*: S is composed of the M subsequences A_i $(i = 1, \ldots M)$ that are concatenated horizontally. Here, we use PSO method or GA to find the optimal subsequences positions in the proposed long sequence, which can be represented by the position affectation vector $P = [P_1, \ldots, P_M]$. Here, P_j $(j = 1, \ldots M)$ is an integer such that $1 \leq P_j \leq M$; it means that the A_{P_j} subsequence is affected to the jth position in the longue sequence S. For example, for $M = 1$, if the metaheuristic algorithm gives $P = [3,1,2]$, then the concatenated subsequences A_3, A_1, and A_2 are affected, respectively, to the first, second, and third positions in the long concatenated sequence S, which is constructed as shown in Figure 3.

2. *Second type (interlacement)*: Here, for M subsequences of n bits, the proposed long sequence S is constructed from the concatenation of n sets of M bits, where the kth set $(k = 1,\ldots,n)$ is obtained by interlacing the kth bits of the M subsequences. The optimal positions of each bit B_i $(i = 1, \ldots M)$ in any set are found using PSO method or GA in the same way as the previous concatenation case. For example, for $M = 3$, if the metaheuristic algorithm gives $P = [3,1,2]$, then the sequence S is constructed as shown in Figure 4.

3 | FITNESS FUNCTION

The measurement of the quality of the optimized sequence S of length N is generally based on the study of its numerical ACF, which is defined as follows:

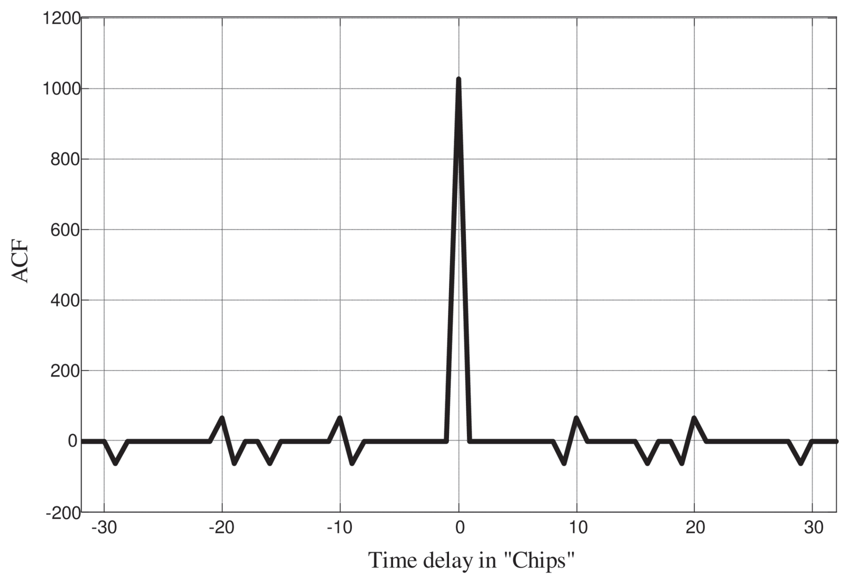


FIGURE 2 Zoomed ACF of a Gold sequence of length 1023 bits

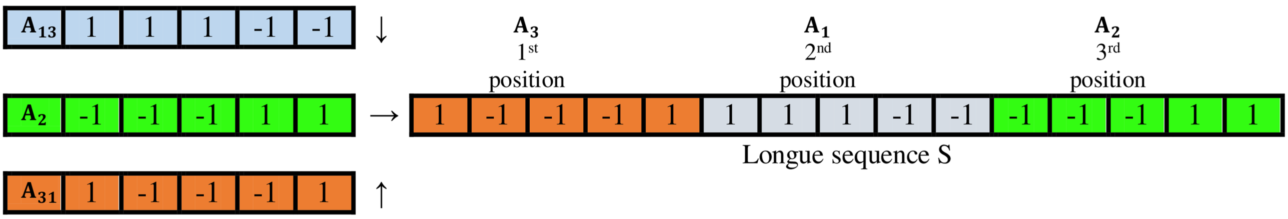


FIGURE 3 Principle of generating the first type of long sequences for $M = 3$ with $P = [3,1,2]$

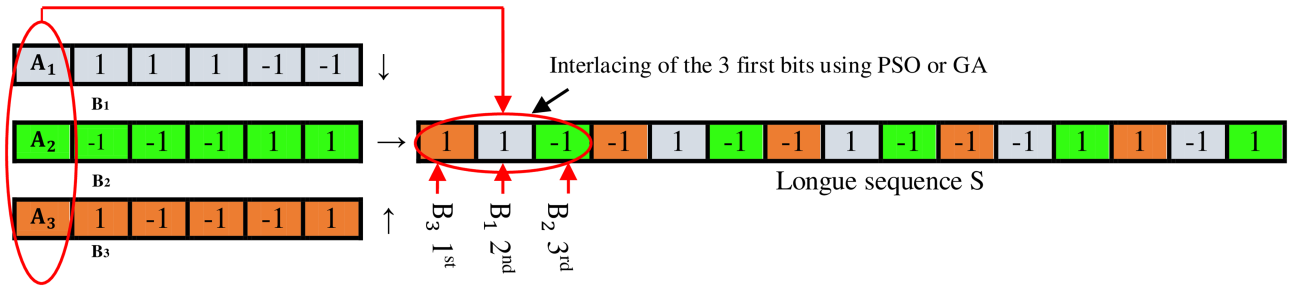


FIGURE 4 Principle of generating the second type of long sequences for $M = 3$ with $P = [3,1,2]$

$$R_S(u) = \sum_{i=1}^{N} S(i)\, S(i-u) \tag{5}$$

Formula 5 assumes that the code bits are expressed as cyclic sequences, with each bit represented as +1 or −1 as appropriate.

In fact, low ACF binary sequences[75,76] are very useful for applications such as channel estimation, radar signal processing, and DSSS communications.

Another important quantitative measure of a code sequence's quality is the cost function or the MF F defined by Golay.[77] It characterizes the difference between the desired and actual ACF properties of a long binary sequence S. It is given, for a binary sequence S of length N, as follows:

$$F(S) = \frac{R_S(0)^2}{\sum_{u \neq 0} |R_S(u)|^2} = \frac{N^2}{2\sum_{u=1}^{N-1} |R_S(u)|^2} \tag{6}$$

where $|R_S(u)|^2$ is the energy of the uth peak and $R_S(0)$ is the central peak value of the ACF.

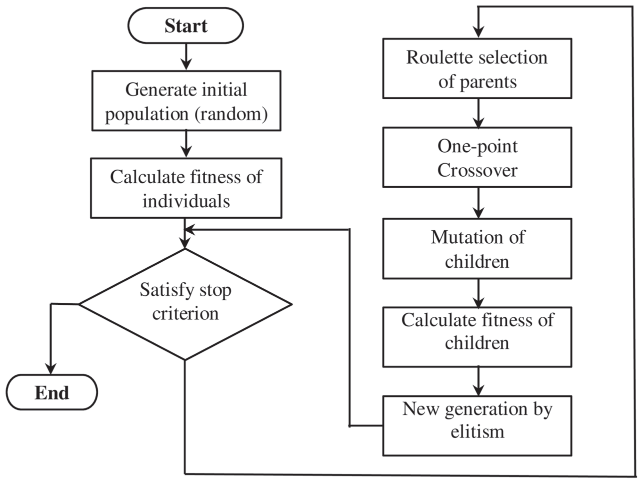
The best binary sequences are those having the largest MF. Therefore, we should minimize the function given by the following:

$$F = \frac{1}{F(S)} \tag{7}$$

4 | LONG SEQUENCES OPTIMIZATION USING GA

The GA is a heuristic search/optimization technique, based on Darwin's theory of evolution "survival of the fittest natural selection," developed by Holland in 1975.[71] This algorithm simulates the processes observed in a natural system where the strong tends to adapt and survive while the weak tends to perish to obtain the best possible solution in a huge solution space. Figure 5 summarizes the GA workflow.

FIGURE 5 Flowchart of the GA workflow



In the encoding and initialization stage, the parameters to be optimized should be encoded into genes and chromosomes. Depending on the type of long binary sequences (concatenated or interlaced), a chromosome is represented by the vector P of M variables, which represents the values of the M subsequences positions or the M bits positions P_i (in the OLBSS), respectively. Thus, in general, a chromosome may be represented as follows:

$$Chromosome = P = [P_1, P_2, P_3, \ldots, P_M] \tag{8}$$

The GA algorithm determines first randomly an initial population of *Npop* chromosomes to be optimized (usually we take $Npop = 40$). Thus, the initial population is a matrix of uniform random numbers given as follows:

$$InitPop = round\left(rand\left(N_{pop}, M\right)\right) \tag{9}$$

Then, the cost function is calculated. This latter is given by the following:

$$Cost = F\,(chromosome) = F\,(P_1, P_2, P_3, \ldots, P_M) \tag{10}$$

In the selection phase, the GA chooses chromosomes for reproduction and how many offsprings each selected individual produces.[78] Among the different selection strategies, we chose the Roulette Wheel, which gives the best results.

The crossover step combines the bits of one chromosome with those of another one.[79] Among the many existing types of crossover, we used the simplest one that is the "single-point" crossover, where two individuals swap their genes at one crossing point.[79] This results in two offsprings, each carrying some genetic information from both parents.

Finally, the mutation operator randomly alters one or more gene values in a chromosome from its initial state. However, with crossover and mutation taking place, there is a high risk that the optimum solution could be lost as there is no guarantee that these operators will preserve the fittest string. To counteract this, elitist preservation is used to prevent the best individuals from going through the reproductive process by passing them unmodified to the next generation.

Two maximum numbers of iterations have been kept in our work: "$MaxIt = 1000$" and "$MaxIt = 2000$" depending on the length of the optimized sequence. In effect, after a certain number of iterations (usually < 1000 *or* < 2000), the sequence stays unchanged which makes it the optimum one. This latter is then memorized, and the algorithm is stopped. The parameters setting of the GA algorithm are given in Table 1.

5 | LONG SEQUENCES OPTIMIZATION USING PSO METHOD

PSO is an optimization technique introduced by Kennedy and Eberhart in 1995.[73,80] It uses a simple mechanism that mimics swarm behavior in birds flocking and fish schooling to guide the particles to search for globally optimal

Population size	***40***
Max number of iterations	*1000/2000*
Crossover	Single point
Crossover rates	*1*
Mutation	Uniform
Mutation rate	*0.02*

TABLE 1 Parameters setting for GA

solutions. Here, each particle is embedded with the relevant information regarding the decision variables and a fitness value that gives an indication of the performance of this particle. The trajectory of each particle is updated according to its own flying experience as well as that of the best particle in the swarm. In our work, the PSO algorithm is used to find the best positions P_{A_i} (first type) or the best positions P_{B_i} (second type). The cost function is the same as in GA.

In every iteration, each particle is updated by following two best position values. The first one is called the personal best position "*pbest*" yielding the best position ever reached by the particle so far and, therefore, the highest fitness value for that particle. The second, which maintains just one best solution, is called the global best "*gbest*" obtained so far by any particle in the population. When a particle takes part of the population as its topological neighbors, the best value is called the local best "*lbest*."

The elements of the implemented PSO algorithm may be given as follows:

i. *Initialization*

In the considered population, where $c = 40$ particles, the ith particle position is represented by the ith subsequence position P_{A_i} for the first type or by the ith bit position P_{B_i} for the second type ($i = 1, \ldots, N_{pop}$). The particle velocities and positions are randomly initialized according to the following steps:

ii. *Calculate the fitness value*

The fitness value of each particle is calculated by the same formula as in GA.

iii. *Update the particle local best*

For each particle, compare the current fitness value with the local optimal one. If current value is better, then replace its optimal fitness value with the current one.

iv. *Update the global best*

For each particle, if the current fitness value is better than the global fitness value, then replace the global best with the current index in the particle array.

v. *Update the velocity and position of particles*

At iteration t, after finding the two best values, the particle updates its velocity and positions with the following equations:

$$v_{ij}^{t} = \chi \left[w * v_{ij}^{t-1} + c_1 r_1 \left(pbest_{ij}^{t-1} - x_{ij}^{t-1} \right) + c_2 r_2 \left(gbest_{ij}^{t-1} - x_{ij}^{t-1} \right) \right] \tag{11}$$

$$x_{ij}^{t} = x_{ij}^{t-1} + v_{ij}^{t} \tag{12}$$

where w is the inertia weight; v_{ij}^{t-1} is the velocity of particle i in the jth dimension at iteration $\mathbf{t-1}$; c_1 and c_2 are the acceleration learning factors; r_1 and r_2 are uniformly generated random numbers in the range [0, 1]; $pbest_{ij}^{t-1}$ is the best solution of particle i in the jth dimension up to iteration $t-1$; $gbest_{ij}^{t-1}$ is the best solution among all the particles up to iteration $t-1$; x_{ij}^{t-1} is the position of particle i in the jth dimension at iteration $t-1$; and finally, χ is the convergence factor, given by the following:

$$\chi = (2k)/|2-\phi-\surd(\phi\hat{}2-4\phi)| \tag{13}$$

where $\phi = \phi_1+\phi_2 \geq 4$; ϕ_1, ϕ_2 and k are convergence factors, with $k \in [0,1]$.

Parameters setting for PSO method are given in Table 2.

6 | PROOF OF THE PROPOSED METHOD

In what fellows, knowing that Walsh–Hadamard subsequences are among the simplest orthogonal structures to be constructed, let's assume that we are interested in realizing an optimized long sequence from a set of 8 Hadamard subsequences of 8 elements each. The initial longue sequence is given by the following vector:

$$\begin{array}{l} S_L = [p\ -p\ p\ -p\ p\ -p\ p\ -p\ p\ p\ p\ -p\ -p\ p\ p\ -p\ -p \cdots \cdot p\ p\ p\ p\ -p\ -p\ -p\ -p] \\ \quad 1^{st}\text{subsequence}\text{"}S_{L1}\text{"} \quad 2^{nd}\text{subsequence}\text{"}S_{L2}\text{"} \cdots\cdots \quad 8^{th}\text{subsequence}\text{"}S_{L8}\text{"} \end{array} \tag{14}$$

In this equation, "$\boldsymbol{p}$" represents the bit positive polarity and "$\boldsymbol{-p}$" the negative bit polarity.

Let us first redefine the correlation function and the orthogonality criteria. Let Had_1 and Had_2 be two distinct Hadamard subsequences, then their numerical ACF and CCF are, respectively, given as follows:

$$ACF_{Had_1}(m) = \sum_{n=0}^{N-1} Had_1(n)Had_1(n-m) \tag{15}$$

$$ACF_{Had_2}(m) = \sum_{n=0}^{N-1} Had_2(n)Had_2(n-m) \tag{16}$$

$$CCF_{Had_1Had_2}(m) = \sum_{n=0}^{N-1} Had_1(n)Had_2(n-m) \tag{17}$$

So that the two sequences Had_1 and Had_2 are orthogonal, they must satisfy the condition given by the following:

$$\sum_{n=0}^{N-1} Had_1(n)Had_2(n) = 0 \tag{18}$$

TABLE 2 Parameters setting for PSO method

Population size	***40***
Number max of iterations	*1000*
Maximum velocity	*30*
Acceleration factors (c_1, c_2)	*1*
Convergence factors	$K = 1$, $\phi_1 = \phi_2 = 2.05$

Now, if we consider the concatenation case, the construction of the proposed long sequence is accomplished by optimizing the subsequences positions in the initial long sequence of (14) using GA or PSO method. This optimization is mainly based on the search of optimal minimization of the lateral peaks' amplitudes on either side of the central peak of the initial long sequence ACF. The mth lateral peak amplitude (from the central peak) is obtained by shifting S_L circularly by m, on the right ($m > 0$) or on the left ($m < 0$), making the products, element by element, between S_L and its shifted version and then summing these products. Now let's consider the cases that correspond to interesting values of the shift value m.

a. **<u>First case</u> "for $m = 0$":**

In this case, as expected, we obtain the maximum value of the ACF corresponding to the central peak amplitude (i.e., the maximum energy) that is equal to 64 (8×8).

b. **<u>Second case</u> "for m a multiple of the length of the Hadamard subsequence" (i.e., m is a multiple of 8):**

In this case, the ACF value is equal to zero. This is simply due to the orthogonality property of the Hadamard subsequences illustrated in Equation 18.

c. **<u>Third case</u> "for $m > 0$ and m is not a multiple of 8":**

Note that the shift values from $m = 1$ to $m = 63$ correspond, respectively, to the lateral peaks from the first to the 63rd whose amplitudes may be calculated as illustrated in the following example.

Let's take $m = 3$; the circularly shifted version of S_L is given as follows:

$$\begin{array}{l} SS_L = [-p\ -p\ -p\ p - p\ p\ -p\ p / -p\ p - p\ p\ p - p - p\ p / p - p\ -p \ldots p\ p\ p\ p - p] \\ \quad 1^{st}\ \text{“8 elements”}(SS_{L1}) \qquad 2^{nd}\ \text{“8 elements”}(SS_{L2}) \end{array} \tag{19}$$

Now if we make the product, element by element, of S_{L1}, and its shifted version SS_{L1}, as depicted in Equations 20 and 21, respectively, we get the result given in Equation 22:

$$S_{L1} = [\boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}] \tag{20}$$

$$SS_{L1} = [-\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}] \tag{21}$$

$$S_{L1} \times SS_{L1} = [-\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}] \tag{22}$$

The shifted sequence in 21 consists of two sets of elements; the first set corresponds to the last three elements of S_{L8}, and the second one corresponds to the five first elements of S_{L1}.

In the resulting products vector, given by 22, we have seven elements of negative polarities and only one element of positive polarity. Hence, the sum of these elements gives the value "−6" which is very close (in absolute value) to the size of the Hadamard subsequence (8) and therefore undesired.

Then, if we make the product, element by element, of S_{L2}, and its shifted version SS_{L2}, given, respectively, by Equations 23 and 24, we get the resulting product vector given in Equation 25, where the numbers of negative and positive polarities are equal. Therefore, the sum of the vector elements is zero, which is very beneficial for the optimization of our long sequence.

$$S_{L2} = [\ \boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}] \tag{23}$$

$$SS_{L2} = [-\boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}\quad \boldsymbol{p}\ -\boldsymbol{p}\ -\boldsymbol{p}\quad \boldsymbol{p}] \tag{24}$$

$$S_{L2} \times SS_{L2} = [-\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \quad -\boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p}] \tag{25}$$

d. **<u>Fourth case</u> "for $m < 0$ and m is not a multiple of 8":**

Let's take $m = -2$; the circularly shifted version of S_L is given as follows:

$$SS_L = [\underbrace{p-p\,p\,p-p-p\,p\,p}_{SS_{L1}}/-p\;-p.....\underbrace{p\,p\,p\,p-p-p}_{SS_{L8}}/-p-p\,p-p\,p-p\,p-p] \tag{26}$$

Using the same reasoning as before, S_{L1}, SS_{L1}, S_{L8},and SS_{L8} are, respectively, given by the following Equations 27–30:

$$S_{L1} = [\boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p}] \tag{27}$$

$$SS_{L1} = [\boldsymbol{p}-\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p}] \tag{28}$$

$$S_{L8} = [\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p}-\boldsymbol{p}-\boldsymbol{p}-\boldsymbol{p}-\boldsymbol{p}] \tag{29}$$

$$SS_{L8} = [-\boldsymbol{p}-\boldsymbol{p} \quad \boldsymbol{p}-\boldsymbol{p} \quad \boldsymbol{p} \;\; -\boldsymbol{p} \quad \boldsymbol{p}-\boldsymbol{p}] \tag{30}$$

Then the following products' vectors are derived:

$$S_{L1} \times SS_{L1} = [\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p}] \tag{31}$$

$$S_{L8} \times SS_{L8} = [-\boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p} \; -\boldsymbol{p} \quad \boldsymbol{p}] \tag{32}$$

The sums of the vectors' elements, given by 31 and 32 are, respectively, equal to "2" and "−2." Therefore, the obtained two values are very beneficial for the optimization of our long sequence. From the presented examples, we see that, for some values of the shift m, we could find small and even null values of the sums of products between the subsequences of S_L and those of its shifted version SS_L. Now, if, for each m value, we do the same thing for each couple of subsequences S_{Li} and SS_{Li} ($1 \le i \le 8$), then the quality of S_L will obviously depend on the results of the products $S_{L1} \times SS_{L1}$, $S_{L2} \times SS_{L2}$,...and $S_{L8} \times SS_{L8}$ for all m values. Moreover, since each SS_{Li} ($1 \le i \le 8$) is composed of two adjacent subsequences, the value of $S_{Li} \times SS_{Li}$ ($1 \le i \le 8$) changes whenever the arrangement of the 8 subsequences is modified. Hence, to improve the overall quality of the initial long sequence S_L, we must test as many different arrangements of subsequences as possible and choose those which insure the lowest sidepeak values and, thus, the most suitable. This optimization process is accomplished by the use of the AG or the PSO method.

7 | SIMULATION RESULTS

Simulations are conducted to test the performances of the OLBSS. For this reason, five scenarios were considered.

In the first scenario, we construct a new class of OLBSS of length 1024 bits composed of the concatenation (first type) or interlacing (second type) of 32 shorter binary Walsh–Hadamard subsequences of 32 bits each. Knowing that Gold sequence has better ACF properties compared to Walsh–Hadamard, the OLBSS is then compared with a traditional Gold sequence having the same length. Here, both GA and PSO optimization techniques are used separately. The simulation parameters, for GA and PSO methods, are given in Tables 1 and 2.

Figure 6 shows the evolution of the fitness function in the first scenario for OLBSS of length 1024 bits in the four different construction cases resulting from the combinations of interlaced/concatenated subsequences and GA/PSO methods. From this figure, it can be observed that the fitness functions' values corresponding to GA are lower than those corresponding to PSO for all iteration numbers allowing, thus, OLBSS with better performances. Besides, the best fitness function value in the case of the GA/interlaced subsequences is equal to 0.8145 at the 854th iteration, while that in the case of the GA/concatenated subsequences is equal to 0.8811 at the 856th iteration. Hence, the OLBSS obtained from the GA/interlaced subsequences gives the best performances compared to that obtained from the GA/concatenated subsequences, the PSO/interlaced subsequences, and the PSO/concatenated subsequences. However, among the two PSO fitness functions, presented in Figure 6, similarly to GA case, it is also that corresponding to interlaced subsequences which exhibits the lowest best fitness function value of 0.9243 at the 942th iteration and allows, thus, OLBSSs with better performances.

To confirm the performance superiority of the best OLBSS composed by Walsh–Hadamard subsequences, as deduced from Figure 6, its ACF is compared to those of the Gold code of same length and the proposed initial long sequence (before optimization).

In Figure 8, we compare the histogram of the non-central-peak ACF values of the best OLBSS and that of non-central-peak ACF values of a random sequence.

Effectively, from Figure 7, it can be observed that the best OLBSS, composed of Walsh–Hadamard subsequences, presents the best ACF characteristics with respect to the original Gold sequence because it exhibits significant reduction in the number of sidepeaks having high levels. In other words, the majority of the best OLBSS sidepeaks with a level above 60 are completely eliminated. This result can be easily explained from Figure 8 where the histogram of the non-central-peak ACF values of random sequence is more spread out than that of the best OLBSS.

In the second scenario, the same previous experiment is repeated but this time with the use of subsequences of same length that belong to the Gold family. Hence, these later are concatenated or interlaced to construct a long sequence that is optimized by using GA or PSO method. The fitness functions are derived for the same combinations as those of the first scenario. Here, the simulation parameters are the same as those given in Tables 1 and 2.

A number of 33 Gold subsequences, of 31 bits length each, are used herein to generate a long sequence of 1023-bit length. Figure 9 illustrates the comparison of the evolutions of the fitness functions given by the different algorithms during the optimization process.

As shown in this figure, it can be observed that the best fitness function value corresponds to the GA/concatenated subsequences and is equal to 1.051 at the 836th iteration. Compared to the Walsh–Hadamard case, as illustrated in Figure 10, this value corresponds to an increase of 28.67%. This is because the Walsh–Hadamard subsequences present a better performance in terms of CCFs that have a direct influence on the constructed OLBSS.

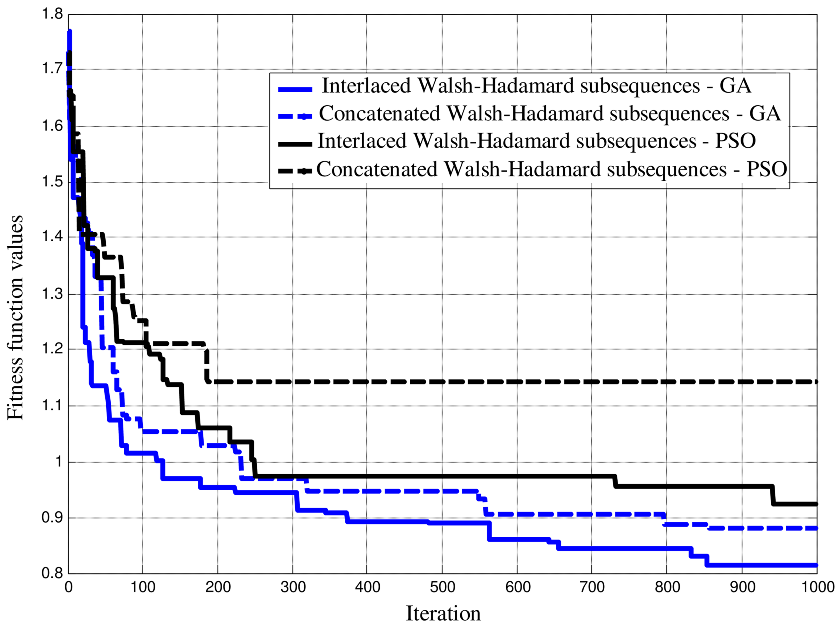


FIGURE 6 Fitness functions for GA/PSO-based OLBSS of 1024 bits long composed of 32 (concatenated/interlaced) Walsh–Hadamard subsequences of 32 bits long

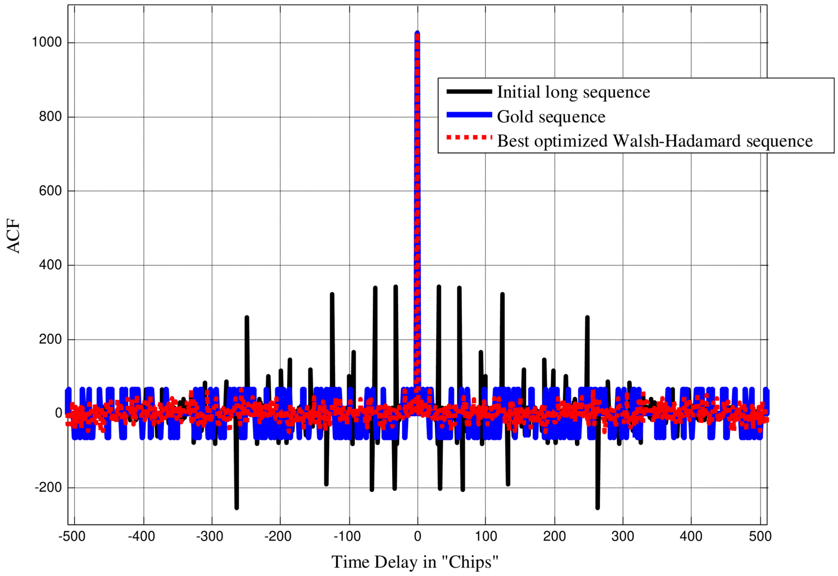


FIGURE 7 Comparison between the ACF of the best OLBSS composed by Walsh–Hadamard subsequences and those of the Gold sequence and the proposed long sequence before optimization

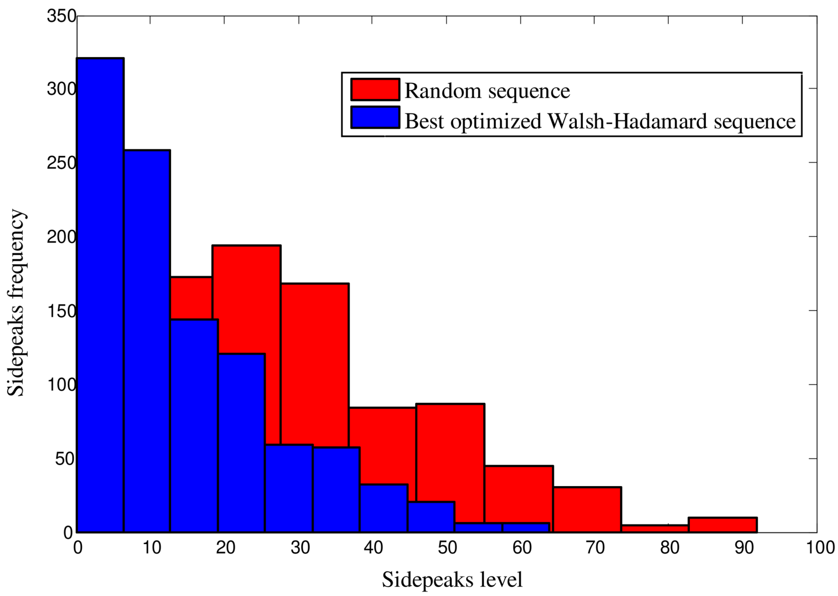


FIGURE 8 Comparison between the histogram of the non-central-peak ACF of the best OLBSS composed of Walsh–Hadamard subsequences and that of non-central-peak ACF values of a random sequence

In the third scenario, we study the effect of the number of subsequences used to construct the OLBSS. For this reason, we construct two different configurations of best OLBSS with 4096-bit length each that are derived from GA/interlaced Walsh–Hadamard subsequences. The first configuration uses 32 subsequences of 128 bits long each, while the second one uses 64 subsequences of 64 bits long each. The evolution of the fitness function corresponding to each case is illustrated in Figure 11.

It is observed from Figure 11 that for a given fixed best OLBSS length, when the number of subsequences is augmented by a half, there is a reduction of about 51.87% of the fitness function values, and thus, the best OLBSS performances become superior in terms of reduction of the sidepeaks levels. Accordingly, the subsequences length should be regarded as an interesting parameter whose judicious choice may improve the best OLBSS quality.

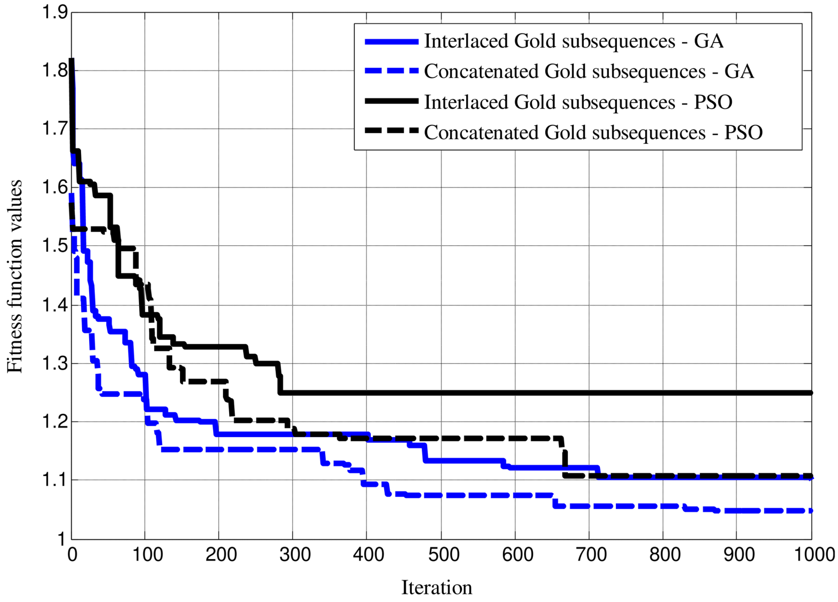


FIGURE 9 Fitness functions for GA/PSO based OLBSS of 1023 bits long composed of 33 (concatenated/interlaced) Gold subsequences of 31 bits long

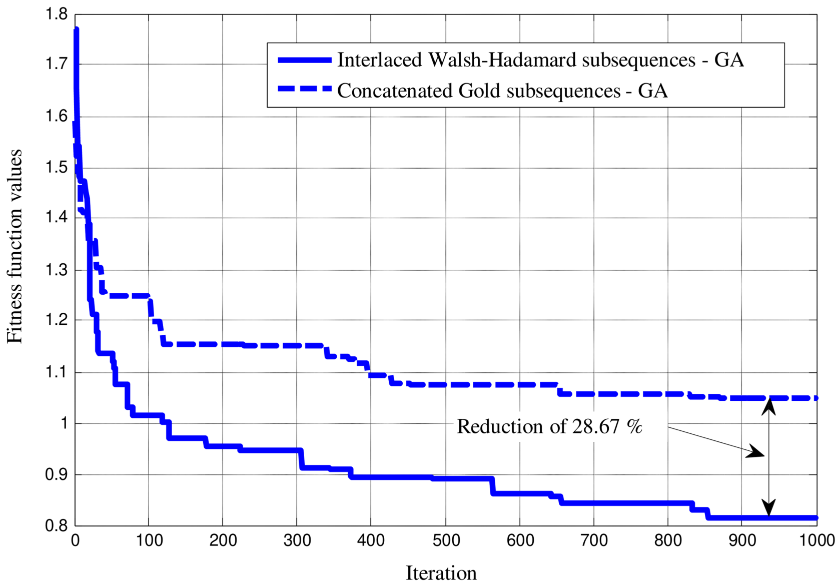


FIGURE 10 Comparison between the fitness functions of the best OLBSS corresponding to GA/interlaced Walsh–Hadamard and GA/concatenated-Gold

In addition to generating OLBSS of medium length as done above, we can also use our proposed method, as illustrated in the following scenario, to generate and optimize longer sequences with better performances. In this scenario, we use the interlacement process of shorter Walsh–Hadamard subsequences in conjunction with the GA to generate a longer OLBSS of size 8192 bits. The comparison of the ACF of the so obtained best OLBSS with those of the random and Gold sequences of the same length is illustrated in Figure 12.

As illustrated in this figure, we directly observe the difference. In fact, the OLBSS, obtained from the interlacement of 64 subsequences of length 128 each, presents an ACF with reduced sidepeaks level. This can be confirmed by Figure 13 where the histograms of the random and Gold sequences are more spread out than that of the OLBSS.

Figure 14 illustrates, for the OLBSS of size 8192 bits, the evolution of the fitness function given by the GA during the optimization process. As shown in this figure, it can be observed that the best fitness function value is equal to

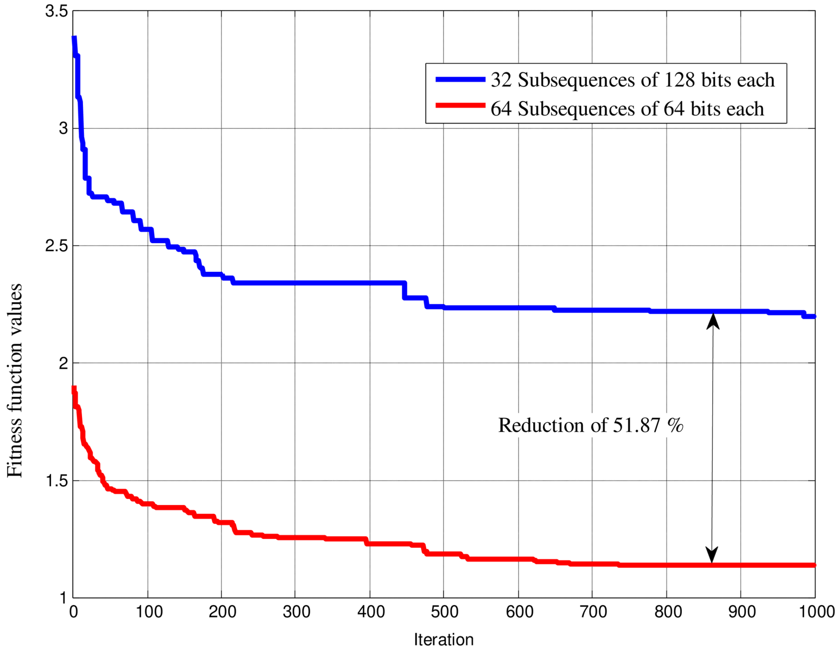


FIGURE 11 Comparison between fitness functions corresponding to two best OLBSS (GA/interlaced Walsh–Hadamard) of 4096 bits length composed, respectively, of 64 subsequences of 64 bits and from 32 subsequences of 128 bits

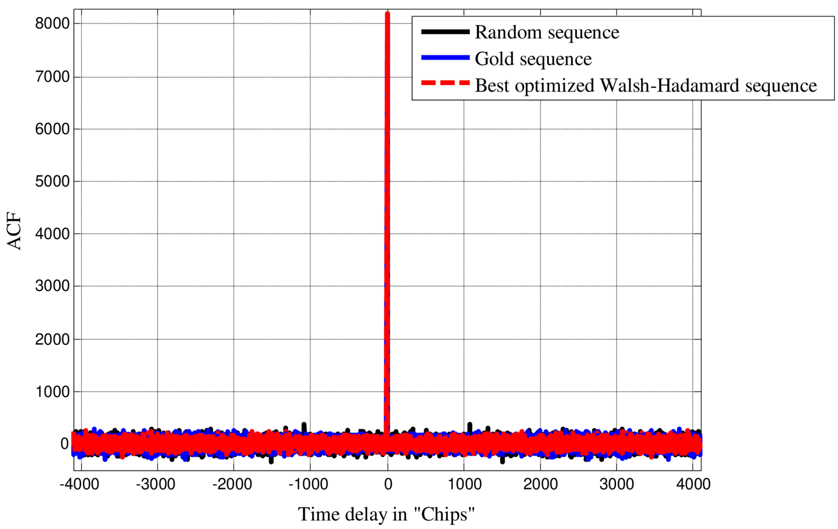


FIGURE 12 Comparison between the ACF of the best OLBSS (GA/interlaced Walsh–Hadamard) of length 8192 bits and those of Gold and random sequences of same length

1.125 at the 1948th iteration. Note here that due to the great length of the obtained sequence, we have used 2000 iterations in the optimization process.

In the last scenario, the efficiency of the proposed method is tested when varying the length of the generated long sequence. For this purpose, a significant number of different length sequences have been considered. These latter were optimized using Hadamard or Gold subsequences in conjunction with GA. The obtained sequences are then compared, respectively, with random sequences of same sizes. To support our results, some optimized sequences were compared also with their corresponding Gold or Weil sequences. The results of comparisons are shown in Tables 3 and 4. Most of the sequences considered in both tables are used in GNSS applications, and their lengths are usually chosen to be an integer power of two (i.e., 2^n) to facilitate FFT computation. From Table 3, the comparative study, based on the F value (inverse of the figure of merit criterion), shows that all different size random sequences have approximately the same F value

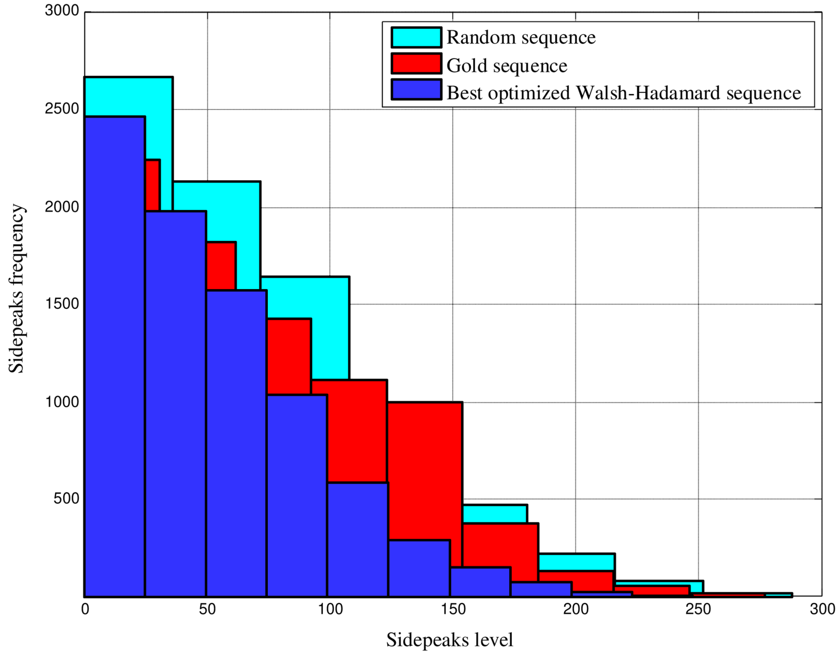


FIGURE 13 Comparison between the histogram of the non-central-peak ACF of the best OLBSS of length 8192 bits composed of Walsh–Hadamard subsequences and those of the non-central-peak ACFs of random and Gold sequences of same length

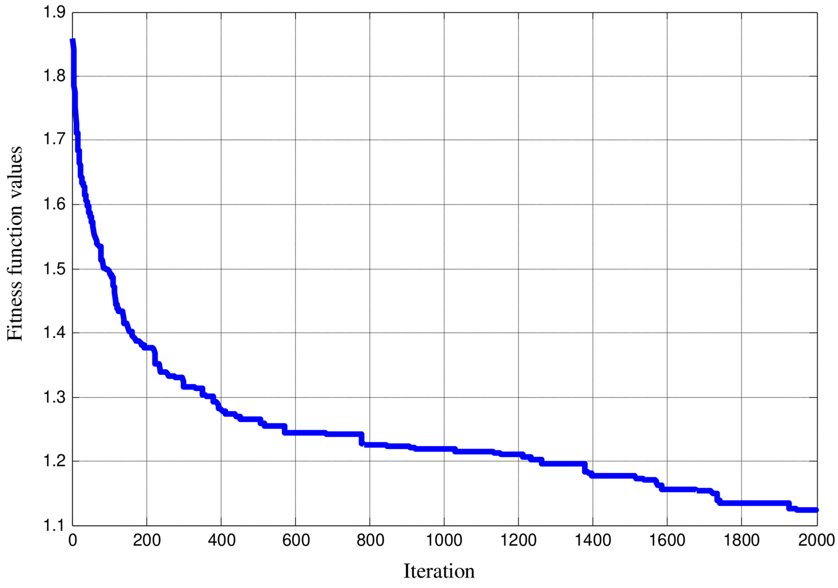


FIGURE 14 Evolution of the GA fitness function for the optimization of a sequence of length 8192 bits composed of 64 interlaced Walsh–Hadamard subsequences of 128 bits each

(around 2), while the optimized sequences, whatever either their sizes, have values that are much smaller. This proves the validity of our proposed method for any sequence length. In fact, the improvement rate, of the optimized sequences, varies from 11.74% (for very long sequences) to 70.37% for short sequences. In addition, contrary to random sequences, the bits +1 and −1 in the proposed sequences (if we exclude the first and the last rows of the Hadamard matrix from the optimization process) are well balanced since the difference between their numbers is zero whatever either their sizes.

From Table 4, the comparative study, based on the same figure of merit criterion, shows that whatever either their sizes, the proposed sequences have values that are much smaller than those of Gold or Weil sequences. In fact, the improvement rate, of the optimized sequences, varies from 22.63% (for very long sequences) to 69.67% for short sequences.

TABLE 3 Comparison of some optimized sequences with random ones at several scales

Proposed sequence length	Subsequence type	Subsequence's number	Subsequence length	*F* value: Proposed OLBSS	*F* value: Random sequence	Improvement rate in %
512	Walsh	16	32	**0.64**	2.16	70.37
1024	Walsh	32	32	**0.81**	2.08	61.06
4096	Walsh	64	64	**0.96**	2.07	53.62
5115	Gold	5	1023	**1.89**	2.10	10.22
10 230	Gold	10	1023	**1.88**	2.13	11.74
16 384	Walsh	128	128	**1.24**	1.99	37.69
65 536	Walsh	256	256	**1.48**	2.01	26.37
262 144	Walsh	512	512	**1.65**	2	17.50
262 143	Gold	513	511	**1.7475**	2.0019	13

TABLE 4 Comparison of some optimized sequences with Gold or Weil sequences at several scales

Proposed OLBSS length	Subsequence type	Subsequence's number	Subsequence length	Traditional sequence type and length	*F* value: Proposed OLBSS	*F* value: Gold Weil sequence	Improvement rate in %
512	Walsh	16	32	Gold/511	**0.64**	2.11	69.67
1023	Gold	33	31	Gold/1023	**1.05**	1.88	44.15
1024	Walsh	32	32	Gold/1023	**0.81**	1.88	56.91
4096	Walsh	64	64	Gold/4095	**0.96**	2.11	54.50
10 230	Gold	10	1023	Weil/10230	**1.88**	2.43	22.63

8 | CONCLUSION

In this paper, an efficient method to generate optimized long spreading sequences with low correlation properties is proposed. The initial long sequence is generated using a number of shorter and same lengths binary subsequences that belong to the family of Walsh–Hadamard or Gold code sequences. At this point, the GA or PSO method is employed to optimize the long sequence ACF characteristics. Herein, the long sequence construction process is done either by concatenation of the subsequences or by their interlacement yielding, therefore, two types of OLBSS. According to the obtained results, the new OLBSS composed of Walsh–Hadamard or Gold sequences have good ACF properties compared to the random and the original Gold sequences having the same lengths. In addition, a comparative study showed that the OLBSS based on GA optimization and interlaced Walsh–Hadamard subsequences presents the overall best performances. Concerning very long OLBSS, the results showed that they still present better performances than those of the random and Gold same lengths' sequences. However, the most efficient among the best OLBSS are those with smaller subsequences number.

ACKNOWLEDGEMENT

The authors would like to thank Pr. Djamel SAIGAA, who has strongly participated in the success of this doctoral training.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Sabrina Boukerma https://orcid.org/0000-0001-8943-876X
Khaled Rouabah https://orcid.org/0000-0001-9238-0554
SalahEddine Mezaache https://orcid.org/0000-0001-7750-6139
Salim Atia https://orcid.org/0000-0002-3756-7276

REFERENCES

1. Sourour E, Atarius R, Khayrallah A. CDMA system using quasi-orthogonal codes. U.S. Patent 7,133,353, 2006.
2. Viterbi AJ. *CDMA Principles of Spread Spectrum*. Addison-Wesley; 1995.
3. Pickholtz R, Schilling D, Milstein L. Theory of spread-spectrum communications—a tutorial. *Navig IEEE Trans Commun*. 1982;30(5):855-884.
4. Sarwate DV, Pursley MB. Correlation properties of pseudo random and related sequences. *Proc IEEE*. 1980;68(5):593-619.
5. Golomb SW. *Shift Register Sequences*. San Francisco: Holden-Day; 1967.
6. Gold R. Optimal binary sequences for spread spectrum multiplexing. *IEEE Trans Inf Theory*. 1967;13(4):619-621.
7. Kasami T. *Weight Distribution Formula for Some Class of Cyclic Codes*. Coordinated Science Laboratory, University of Illinois at Urbana-Champaign; 1966.
8. Rushanan J J. Weil Sequences: A Family of Binary Sequences with Good Correlation Properties IEEE International Symposium on Information Theory, Seattle, WA, USA, 2006.
9. European Space Agency/European GNSS Supervisory Authority. *Galileo Open Service. Signal in Space Interface Document*. OS SISICD; 2008.
10. Sharifi M, Jalali MJP. Using chaotic sequence in direct sequence spread Spectrum based on code division multiple access (DS-CDMA). *ARPN J Eng Appl Sci*. 2017;12(20):5837-5846.
11. Fan PZ, Suehiro N, Kuroyanagi N, Deng XM. Class of binary sequences with zero correlation zone. *Electr Lett*. 1999;35(10):777-779.
12. Liu Y-C, Chen C-W, Su YT. New constructions of zero-correlation zone sequences. *IEEE Trans Inf Theory*. 2013;59(8):4994-5007.
13. Golomb SW. *Shift Register Sequences: Secure and Limited-Access Code Generators, Efficiency Code Generators, Prescribed Property Generators, Mathematical Model. (Third Revised Edition)*. World Scientific; 2017.
14. Rouabah K, Attia S, Harba R, Ravier P, Boukerma S. Optimized method for generating and acquiring GPS Gold codes. *International Journal of Antennas and Propagation*. 2015;2015:1-9.
15. Wallner S. Avila-Rodriguez J-A, Hein G W, Galileo E1 OS and GPS L1C Pseudo Random Noise. Codes - Requirements, Generation, Optimization and Comparison. Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS), 2007.
16. Dinan EH, Jabbari B. Spreading codes for direct sequence CDMA and wideband CDMA cellular networks. *IEEE Commun Mag*. 1998;36(9):48-54.
17. Kedia D, Duhan M, Maskara S L. Evaluation of Correlation Properties of Orthogonal Spreading Codes for CDMA Wireless Mobile Communication. IEEE 2nd International Advance Computing Conference (IACC), 2010, Patiala, India.
18. Adachi F, Sawahashi M, Okawa K. Tree-structured generation of orthogonal spreading codes with different lengths for forward link of DS-CDMA Mobile radio. *Electronics Letters*. 1997;33(1):27-28.
19. ETSI. The ETSI UMTS Terrestrial Radio Access (UTRA) ITU-R R0TT Candidate Submission, 1998.
20. Association of Radio Industries and Businesses (ARIB). *Japan's Proposal for Candidate Radio Transmission Technology on IMT-2000*. W-CDMA; 1998.
21. CATT. TD-SCDMA Radio Transmission Technology for IMT-2000 Candidate submission, 1998.
22. TIA/EIA-95B. Mobile station-base station compatibility standard for dual-mode wideband spread spectrum cellular systems. In: *Baseline Version*; 1997.
23. Telecommunications Industry Association (TIA). *The CDMA2000 ITU-R RTT Candidate Submission*, 1998.
24. Holma H, Toskala A. *WCDMA for UMTS Radio Access for Third Generation Mobile Communications*. 1st ed. John Wiley & Sons; 2000.
25. Stark WE, Sarwate DV. Kronecker Sequences for spread Spectrum communication. *IEE Proc F - Commun, Radar Signal Proc*. 1981;128(2):104-109.
26. Alamgir H, Shariful I, Sadek A. Performance analysis of barker code based on their correlation property in multiuser environment. *Int J Inf Sci Techn (IJIST)*. 2012;2(1):27-39.
27. Golomb S, Scholtz R. Generalized Barker sequences. *IEEE Trans Inf Theory*. 1965;11(4):533-537.
28. Scholtz R, Welch L. GMW sequences. *IEEE Trans Inf Theory*. May 1984;30(3):548-553.
29. Olsen J, Scholtz R, Bent-Function Sequences WL. Bent-function sequences. *IEEE Trans Inf Theory*. 1982;28(6):858-864.
30. No J-S, Kumar PV. A new family of binary pseudorandom sequences having optimal periodic correlation properties and large linear span. *IEEE Trans Inf Theory*. 1989;35(2):371-379.
31. Chen H-H. Multi band wavelet packet spreading codes with intra code sub band diversity for communication in multipath fading channels. *IEICE Trans Commun*. 2001;E84-B(7):1876-1884.
32. Someswar GM, Rao TPSC, Chigurukota DR. Global navigation satellite systems and their applications. *Int J Softw Web Sci (IJSWS)*. 2013;3(1):17-23.

33. Hofmann-Wellenhof B, Lichtenegger H, Wasle E. *GNSS Global Navigation Satellite Systems; GPS, Glonass, Galileo & More*. Springer-Verlag Wien; 2007.
34. Avila-Rodriguez JA. *On Generalized Signal Waveforms for Satellite Navigation*. Universitätsbibliothek der Universität der Bundeswehr München; 2008.
35. Wallner S, Avila-Rodriguez JA, Hein GW. *Codes: The PRN Family Grows Again*. Inside GNSS; 2011.
36. Winkel J O. Spreading Codes for a Satellite Navigation System. U.S. Patent 8,035,555, 2011.
37. Hein GW, Avila-Rodriguez J-A, Wallner S. The ~~DaVinci~~ Galileo code and others. *Inside GNSS*. 2006;1(6):62-74.
38. Kaplan ED, Understanding HCJ. *GPS: Principles and Applications*. 2nd ed. Boston: Artech House; 2006.
39. Borre K, Akos DM, Bertelsen N, Rinder P, Jensen SH. *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*. Birkhäuser: Bosten; 2007.
40. Stansell T A, Hudnut K W, Keegan R G. GPS L1C: Enhanced Performance, Receiver Design Suggestions, and Key Contributions. Proceedings of the 23rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2010), Portland, OR, 2010.
41. Lu H, Niu R. Generation method of GPS L1C codes based on quadratic reciprocity law. *J Syst Eng Electr*. 2013;24(2):189-195.
42. Ding C, Hesseseth T, Shan W. On the linear complexity of legendre sequences. *IEEE Trans Inf Theory*. 1998;44(3):1276-1278.
43. European Union. European GNSS (GALILEO) Open Service Signal-In-Space Interface Control Document. OS SIS ICD, 2016, 1.3.
44. Wallner S, Avila-Rodriguez J-A, Won J-H, Hein G W. Issler J-L. Revised PRN Code Structures for Galileo E1 OS. Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008), Savannah, GA, 2008: 887-899.
45. Gao GX, Chen A, Lo S, de Lorenzo D. *Enge P*. GNSS over China, the Compass MEO satellite codes: Inside GNSS; 2007:36-43.
46. Gao GX, Chen A, Lo S, De Lorenzo D, Walter T, Enge P. Compass-M1 broadcast codes in E2, E5b, and E6 frequency bands. *IEEE J Sel Top Signal Processing*. 2009;3(4):599-612.
47. Russian Space Systems. Global navigation satellite system GLONASS. In: *Interface Control Document*; 1998.
48. Zepeng Z, Jinfeng C, Lei Y, Zhiyao Y. Correlation functions of perfect binary sequences of same lengths. *Wuhan Univ J Nat Sci*. 2019;24(3):201-204.
49. Kedia D. Comparative analysis of peak correlation characteristics of non-orthogonal spreading codes for wireless systems. *Int J Distrib Parallel Syst (IJDPS)*. 2012;3(3):63-74.
50. Ren J. Design of long period pseudo-random sequences from the addition of M -sequences over Fp. *EURASIP J Wirel Commun Netw*. 2004;1:12-18.
51. Sarayloo M, Gambi E, Spinsante S. A new approach to sequence construction with good correlation by particle swarm optimization. *J Commun Softw Syst*. 2015;11(3):127-135.
52. De Bruijn NG. A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen te Amsterdam*. 1946;49(2):758-764.
53. Tan L, Xu H, Qi W-F. Preliminary results on the minimal polynomial of modified de Bruijn Sequences. *Finite Fields Appl*. 2018;50:356-365.
54. Alhakim A, Akinwande MA. Recursive construction of nonbinary de Bruijn sequences. Designs, codes and cryptography. *Springer*. 2011;60(2):155-169.
55. Li C, Zeng X, Helleseth T, Li M. Construction of De Bruijn sequences from LFSRS with reducible characteristic polynomials. *IEEE Trans Inf Theory*. 2016;62(1):610-624.
56. Dong J, Pei D. Construction for de Bruijn sequences with large stage. Designs, codes and cryptography. *Springer*. 2017;85(2):343-358.
57. Chan AH, Games RA, Key EL. On the complexities of De Bruijn sequences. *J Comb Theory Ser a*. 1982;33(3):233-246.
58. Etzion T. Linear complexity of De Bruijn Sequences—old and new results. *IEEE Trans Inf Theory*. 1999;45(2):693-698.
59. Meng R, Yan T. New constructions of binary interleaved sequences with low autocorrelation. *Int J Netw Secur*. 2017;19(4):546-550.
60. Zhao L, Song J, Babu P, Palomar DP. A unified framework for low autocorrelation sequence design via majorization-minimization. *IEEE Trans Signal*. 2017;65(2):438-453.
61. Du K L, Wu W H, Mow W H. Determination of long binary sequences having low autocorrelation functions. U.S. Patent 2012/0062399, 2012.
62. Leukhin AN, Parsaev NV, Bezrodnyi VI, Kokovihina NA. The exhaustive search for optimum minimum peak sidelobe binary sequences. Bulletin of the Russian Academy of Sciences. *Phys Ther*. 2017;81(5):575-578.
63. Dimitrov M, Baitcheva T, Nikolov N. Efficient generation of low autocorrelation binary sequences. *IEEE Signal Proc Lett*. 2020;27:341-345.
64. Alaee-Kerahroodi M, Modarres-Hashemi M, Naghsh MM. Designing sets of binary sequences for MIMO radar systems. *IEEE Trans Signal Proc*. 2019;67(13):3347-3360.
65. Bose A, Mohammadi N, Soltanalian M. Designing signals with good correlation and distribution properties. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018.
66. Stoica P, He H, Li J. New algorithms for designing unimodular sequences with good correlation properties. *IEEE Transactions on Signal Processing*. 2009;57(4):1415-1425.
67. Bose A, Soltanalian M. Constructing binary sequences with good correlation properties: an efficient analytical-computational interplay. *IEEE Transactions on Signal Processing*. 2018;66(11):2998-3007.

68. Soltanalian M, Naghsh MM, Stoica PA. Fast algorithm for designing complementary sets of sequences. *Signal Process*. 2013;93(7):2096-2102.
69. Lin R, Soltanalian M, Tang B, Li J. Efficient design of binary sequences with low autocorrelation sidelobes. *IEEE Trans Signal Process*. 2019;67(24):6397-6410.
70. Stoica P, He H, Li J. On designing sequences with impulse-like periodic correlation. *IEEE Signal Process Lett*. 2009;16(8):703-706.
71. Holland JH. *Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence. The*. University of Michigan Press; 1975.
72. Goldberg DE. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison Wesley; 1989.
73. Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, Perth, WA, Australia, Australia, 1995.
74. Eberhart, Shi Y. Particle swarm optimization: developments, applications and resources. Proceedings of the Congress on Evolutionary Computation; Seoul, South Korea, South Korea, 2001.
75. Bernasconi J. Low autocorrelation binary sequences: statistical mechanics and configuration space analysis. *J Phys*. 1987;48(4):559-567.
76. Mertens S. Exhaustive search for low-autocorrelation binary sequences. *J Phys a Math Gen*. 1996;29(18):L473-L481.
77. Golay MA. Class of finite binary sequences with alternate autocorrelation values equal to zero. *IEEE Transactions on Information Theory*. 1972;18(3):449-450.
78. Blickle T, Thiele LA. Comparison of selection schemes used in genetic algorithms. *Evol Comput*. 1996;4(4):361-394.
79. Mitchell M. *An Introduction to Genetic Algorithms*. Bradford Books; 1998.
80. Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput*. 2002;6(1):58-73.

How to cite this article: Boukerma S, Rouabah K, Mezaache S, Atia S. Efficient method for constructing optimized long binary spreading sequences. *Int J Commun Syst*. 2021;34:e4719. https://doi.org/10.1002/dac.4719