

A New Approach to Sequence Construction With Good Correlation by Particle Swarm Optimization

Mahdiyaz Sarayloo, Ennio Gambi, and Susanna Spinsante

Original scientific paper

Abstract—In this paper, a novel computationally affordable method to generate long binary sequences featuring desired properties is presented, based on the use of a number of shorter non linear binary sub-sequences. The paper shows the relationship of the Auto- and Cross-Correlation (AC, CC) of the generated long binary sequences with the AC and CC of constituent sub-sequences. It is also shown that the starting bit position of sub-sequences has an important role on AC and CC of the generated sequences. To generate the optimal long binary sequence from correlation points of view, Particle Swarm Optimization (PSO) algorithm is employed. All the techniques stated in the literature to improve the PSO are implemented and it is clearly shown that the constriction factor and the variable population size turn out to have a great impact on minimizing the fitness function (RMS of AC) representing the target Correlation properties expected for the resulting long sequence. Possible application scenarios for the long sequences generated by the proposed method are also discussed and evaluated.

Index Terms—Auto-Correlation; Cross-Correlation; Particle Swarm Optimization; De Bruijn Sequences

I. INTRODUCTION

DIRECT Sequence Spread Spectrum (DS-SS) technologies differentiate the users' channels by codes rather than by frequency filtering and time gating, thereby allowing for multiple access on the same channel throughout the network. In SS technology, the signal transmission is spread over a broad band; the spreading scheme is known only to authorized users, while for unauthorized ones the signal is embedded in noise and undetectable. This technology found first wide usage in military communications, as a Low Probability of Intercept (LPI) and anti-jamming method. Later on, it was applied in the civil communications scenario, to enable, among the others, multiple users communications. Code Division Multiple Access (CDMA) was first proposed to the cellular service in 1989, and commercially introduced in the mid 90s.

Since then, CDMA has been employed in many radar and satellite-based communication systems, such as Globalstar [1], and even in Intelligent Transport System (ITS) [2]. In the satellite domain, CDMA is employed in some of the Global Navigation Satellite Systems (GNSS) like Galileo, COMPASS (provided by China) and Global Positioning System (GPS). It is highly important to note that conversion of FDMA to CDMA in GLONASS is reported as an increasing likely event [3], [4]. Multi Carrier CDMA (MC-CDMA) is a multiple

access (MA) technique which combines the DS-CDMA and the orthogonal frequency division multiplexing (OFDM) signaling. The robustness of MC-CDMA to multipath fading and the low chip rate requirement are among the most important advantages provided by this technique [5]. Wideband CDMA (WCDMA) is another type of CDMA used in the third generation (3G) mobile systems which is designed for multimedia communications with higher data rate [6]. Frequency Division Duplex (FDD) and Time Division Duplex (TDD) operations are covered by WCDMA. The Pseudo Noise (PN) sequences are being used for both purposes of spreading and scrambling the signal, to expand its bandwidth and randomize the spread signal, respectively [7].

Different parameters take effect on the performance of a DS-CDMA system. Correlation properties are among the most significant features of the PN sequences used as spreading codes, that are to be evaluated. The similarity of two signals may be determined by this measure: the correlation between two different sequences is called Cross-Correlation (CC), otherwise, Auto-Correlation (AC). The AC profile may impact on the interference mitigation and subsequently on the CDMA capacity [8], [9]. The ideal AC profile is a Dirac delta function which is maximum at the main-lobe (null shift) and has a 0 value at any other position (the so-called side-lobes).

To emphasize the impact of the sequence correlation properties on the spreading system, a few cases can be mentioned as follows. In MC-CDMA, by maximizing the AC of the received signal, the number of subcarriers and the guard interval are primarily determined; secondly, the distortion on the received signal is degraded [10]. Many techniques may be found in the literature to overcome the different types of interference. The use of spreading sequences with appropriate correlation properties may allow to effectively control the multiple access interference (MAI) and inter-symbol interference (ISI) [11]. The correlation properties of the sequence set also impact on the capacity of both CDMA [12] and optical CDMA [13] systems. Sequences with low CC properties are highly demanded in MA systems. Further, since the scrambling codes are employed to randomize the users, good correlation properties are also required for these codes [14], [15]. A unique scrambling code is utilized in each WCDMA cell [16].

Most of the widespread used binary sequences are linear, meaning that they can be generated through linear functions, such as additions and multiplications, often by means of deterministic constructions such as Linear Feedback Shift Registers (LFSRs). However, adopting non-linear binary sequences may

Manuscript received June 23, 2015; revised September 11, 2015.
Authors are with the Department of Information Engineering, Polytechnic University of Ancona, Italy (Email: {m.sarayloo,e.gambi,s.spinsante}@univpm.it).

provide further benefits, such as increased robustness against attacks trying to guess the spreading code used, increased cardinality of the sequence set, and better randomness properties [17]. The sequence construction method presented in this paper relies on short non-linear De Bruijn binary sequences to generate long non-linear codes. In fact, the generation of non-linear sequences exhibits great complexity, especially when the length of the sequences increases, which makes it not affordable in real-time systems [18]. Longer sequences with desired properties may be achieved by selecting and combining short codes, through suitable optimization algorithms, as discussed within the paper.

Particle Swarm Optimization (PSO) is one of the heuristic search method which is inspired by the ability of the flocks of birds and school of fishes and herds of animal. The information sharing is the main key of their ability to adapt to their environment, avoid the predators and find the source of food [19]. The original PSO (OPSO) is proposed by Kennedy and Eberhart [20]. According to the literature, PSO is more efficient computationally [19] compared to the Genetic Algorithm (GA) and has better performance in terms of success rate and solution quality in comparison with GA, Memetic Algorithms (MA), Shuffled Frog Leaping (SFL) and Ant-Colony Optimization (ACO) [21]. Many researches have been attempted to improve the OPSO by use of different techniques explained in section IV-A

Many researches may be found in the literature dealing with the generation of the spreading sequences with low correlation properties to improve the performance of communication systems. The available works are discussed in II.

The rest of paper is organized as follows: after reviewing related work on binary sequences construction in Section II, the proposed construction method is presented in Section III, where the desired AC properties of the final binary sequences are also discussed. The same Section also motivates the use of De Bruijn Sequences (DBSs) in the generation process. The Particle Swarm Optimization (PSO), applied as an Evolutionary Algorithm (EA) to generate the optimal sequences (i.e. featuring low AC side-lobes), according to the defined constraints, is discussed in Section IV. Section V presents the simulation results on the proposed construction algorithm and on the application of the generated sequences in Long Term Evolution (LTE) systems; finally, Section VI draws the conclusion of the paper.

II. RELATED WORK

Liu, in [22], proposes a binary sequence construction method based on a chaotic system. Randomness and also correlation properties of the generated sequences are considered in the paper. Despite the generated sequences exhibit good properties, the author does not recommend this method to be used for generating long sequences, in systems operating in real-time. A new method based on generalized orthogonality is presented in [23], to generate the binary sequences with Zero Correlation Zone (ZCZ) of width $Z_0 \leq \frac{N}{|C|} - 1$, where N and $|C|$ are the sequence length and the size of sequences set (also called *cardinality*), respectively. The ZCZ

is a set of shift values for which the sequence correlation amounts to zero. According to the results discussed in this paper, generating sequences featuring a ZCZ longer than the maximum channel delay helps suppressing the ISI and MAI. However, the sequences generated by means of the method proposed in this paper outperform the ones obtained according to [23], from the correlation point of view.

A novel PN sequence generator is presented in [24], in which LFSR-based construction is utilized. The novelty of this implementation is the use of a *dynamic* feedback. After applying the correlation and security analysis, reported in [25], on the sequences generated in [24], they have not been found useful for CDMA communications, due to the poor correlation quality.

To the best of our knowledge, [26] and [27] are the only works that investigate on the generation of binary random sequences based on EA. The use of Genetic Algorithm (GA) is the approach proposed in [26] to generate DBSs. Two types of mutation and also crossover are used in the implemented GA to produce the new children. It must be noticed that the method in [26] does not guarantee to generate the full length DBSs for any span value. In addition, the author denotes the fact that the presented method is not efficient for generating DBSs of span $n > 14$. Even if Turan, in [26], deploys GA as an EA to generate DBSs, the authors have found no work in the literature about the use of EA in the construction of PN binary sequences with desired properties. Mow et al. recently proposed an EA in which GA, Evolutionary Strategy (ES) and memetic algorithm are combined to generate the long binary sequences with low AC [27]. Considering the presented results in [27], the long execution time may be stated as a drawback of this methodology, with respect to the computationally affordable one herein proposed.

Considering the stated literature about the generation of suitable PN sequences for different communication systems working with ranging codes, we aim at investigating on a computationally feasible method, to be executed in real-time, as a new approach to the problem, rather than a pure mathematical one.

III. SEQUENCE CONSTRUCTION AND TARGET PROPERTIES

In this section, the sequence construction method is introduced, and the expected correlation properties of the generated sequences are discussed. Then, the contribution of DBSs and motivation for their adoption are provided.

A. Construction method and AC analysis

The construction method is indeed very simple: the new sequence is composed by two or more different sequences of the same length, i.e. belonging to the same family. Assuming the simple case of two different sequences denoted as $A = \{a_1, a_2, \dots, a_N\}$ and $B = \{b_1, b_2, \dots, b_N\}$, where N is the sequence length, the new sequence would be presented as $S = \{a_1, b_1, a_2, b_2, a_3, b_3, \dots, a_N, b_N\}$.

It is worthwhile to investigate on the AC profile of the aforementioned sequence S , considering the fact that we know the AC profile of all the possible sub-sequences, selected from

the same family. We focus on periodic AC and CC, which are defined as Eq.(1) where C and k represent the correlation property and the number of shift, respectively. If $a = b$, the AC is measured, otherwise the measured property is CC. Let us assume, for the sake of simplicity and without loss of generality, the length of each sequence (A , and B) is 5; hereby, the length of the mixed sequence is 10. The AC of this sequence has one main-lobe (at a zero shift) and nine side-lobes. Further, shifting the sequence to the right or to the left has no effect on the computation of the periodic correlation. Due to this fact, the analysis can be restricted to the first five side-lobes only, the values of which are reported in Table I: $CC_{AB}(k, B^R)$ denotes the CC of sequences A and B when sequence B is shifted to the right (B^R), by k positions.

$$C_{ab}[k] = \frac{1}{N} \sum_{i=0}^{N-1} a(i)b(i+k) \quad (1)$$

TABLE I

AC VALUES OF A MIXED SEQUENCE $S = \{a_1, b_1, a_2, b_2, \dots, a_5, b_5\}$, ACCORDING TO THE SHIFT AMOUNT AND THE AC/CC VALUES OF THE SUB-SEQUENCES A AND B

Shift	Formula
0	$AC_A(0) + AC_B(0)$
1	$CC_{AB}(0, B^R) + CC_{AB}(1, B^R)$
2	$AC_A(1^R) + AC_B(1^R)$
3	$CC_{AB}(4, B^R) + CC_{AB}(2, B^R)$
4	$AC_A(2^R) + AC_B(2^R)$
5	$CC_{AB}(3, B^R) + CC_{AB}(3, B^R)$

The AC of a mixed sequence S may be formulated as follows. In Eq.(2), $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ are the floor and the ceiling functions, respectively. According to Eq.(2), when the shift k is odd, the AC of the mixed sequence S depends on two CC terms computed over the sub-sequences. The interesting point is that the sum of the arguments of the two CC terms equals $2N + 1$. Having a low AC in the resulting sequence S requires to either have low $|CC|$ for the mentioned terms in Eq.(2), or two terms of opposite sign, to suppress the effect of each other.

$$AC(k) = \begin{cases} AC_A(k/2) + AC_B(k/2) & \text{if } k : \text{Even} \\ CC_{AB}(2N - \lfloor k/2 \rfloor, B^R) + CC_{AB}(\lceil k/2 \rceil, B^R) & \text{if } k : \text{Odd} \end{cases} \quad (2)$$

It is encouraged to expand this equation for the general case of a sequence composed by M different sub-sequences

B. Contribution of DBSs

Pseudorandom sequences are typically generated by means of Feedback Shift Registers (FSR), in the sense that each element of the sequence is a function $f(\cdot)$ of the previous n element. If $f(\cdot)$ is a linear function, it is called a Linear FSR (LFSR), otherwise it is a Non-linear FSR (NFSR). Considering the security aspects, and the increasing trend in the number of users to be served by a wireless communication system, predictability and small cardinality may be stated as the two main drawbacks of the sequence sets generated by LFSRs. DBSs are generated based on NFSRs, with high linear complexity [28]. Many approaches are available in the literature,

about the generation of DBSs. Some of them are based on the graph theoretic approach [29], relying on sequences of span n to build new sequences of span $n + 1$ [30]–[32]. It is highly important to emphasize the fact that the generation of full sets of DBSs is still an open issue, for high span values.

The length and the number of distinct sequences are among the most important parameters to evaluate in a sequence family. These two parameters depend on the number of symbols in the alphabet (k) and span value (n), as mentioned in Eq.(3) and Eq.(4) where N and $|C|$ denote the length and the number of distinct sequences [33], respectively. Since, in this paper, it is intended to use binary sequences, these equations would be simplified by substituting $k = 2$. The huge cardinality of DBSs ($|C|$) is an advantage to exploit in multiuser applications with dense scenarios [18].

$$N = k^n \quad (3)$$

$$|C| = \frac{k!k^{n-1}}{k^n} \quad (4)$$

As mentioned in [34], the AC and CC of DBSs are bounded according to Eq.(5) and Eq.(6). The maximum AC, as stated by this equation, is $2^n - 4\lceil \frac{2^n}{2n} \rceil$, which is a considerably high value. Table II shows the normalized maximum absolute values of the AC side-lobes, for DBSs of span values from 3 to 5. As it is clear from the table, the maximum AC side-lobe value increases if the span value increases, which is not desirable. Likewise, according to Eq.(6), CC of DBSs covers a significant range, meaning that DBSs may have a high similarity in some index shifts.

$$0 \leq \max AC(k) \leq 2^n - 4\lceil \frac{2^n}{2n} \rceil, 1 \leq k \leq N - 1 \quad (5)$$

$$-2^n \leq CC(k) \leq 2^n - 4, 0 \leq k \leq N - 1 \quad (6)$$

TABLE II

MAXIMUM NORMALIZED AC SIDE-LOBE VALUES FOR DBSs OF DIFFERENT SPAN

n	$\max(AC(k \neq 0))$
3	0.5
4	0.75
5	0.75

There are two important reasons to motivate the use of DBSs in the aforementioned sequence construction method. First, the generation of DBSs of higher span values is difficult, such that it is still an open issue. Second, the $\max(|AC(k \neq 0)|)$ and $\max(|CC|)$ of DBSs increase proportionally to the span value. In addition, as mentioned in the previous section, DBSs may be started from specific bit positions so that either CCs have low values, or suppress the effect of each other.

It can be argued that the behavior of AC is important for mixing more than two sequences in a proper way. According to the sequence construction method and the derived equation (Eq.(2)), we need two long sequences to generate a single longer sequence of double length. It still may cause problems

when generating long sequences, like 1024 bit long, since two sequences of length 512 are required, with a good AC profile. Therefore, it is intended to derive a general equation to calculate the AC of a sequence composed by M sub-sequences. Eq.(7) shows the AC equation of sequence S obtained by the mix of M sub-sequences, for a shift k :

$$AC_S(k) = \begin{cases} \sum_{i=1}^M CC_{A_i A_{\lambda(j,k)}} (Shf_{in}, A_{\lambda(j,k)}^R) & \text{if } \text{mod}(k, M) \neq 0 \\ \sum_{i=1}^M AC_{A_i}(k/M) & \text{if } \text{mod}(k, M) = 0 \end{cases} \quad (7)$$

In the above equation, l_s is the length of each sub-sequence, and $Shf_{in}(k, i)$ is the shift index that should be applied on a sub-sequence A_{λ} , being $\lambda = \lambda(j, k)$ according to Eq.(9). The cross-correlation terms denoted by CC have two arguments: the former shows the amount of shift, and the latter indicates which sequence should be shifted. Moreover, the superscript of the second argument shows that the sub-sequences are always shifted to right.

To understand how Eq.(7) is derived, three main issues must be paid attention to:

- 1) The number of sub-sequences is presented by M .
- 2) All the sub-sequences are defined as bipolar arrays with the same length l_s . Subsequently, the length of the resulting sequence S is $M \cdot l_s$.
- 3) The sequence obtained mixing the sub-sequences is shown by S and the sub-sequences are illustrated by A_i , where i is the sub-sequence index, from 1 to M . The sequence bits are presented as $a_{i,q}$, where $i = 1 \dots M$ and $q = 1 \dots l_s$.
- 4) In the sequence S , all the sub-sequences are regularly repeated, like $S = \{a_{11}, a_{21}, \dots, a_{M1}, \dots, a_{1l_s}, a_{2l_s}, \dots, a_{Ml_s}\}$ if sequence S is composed by M sub-sequence A_1, A_2 and A_M , of length l_s .

According to the mentioned points above, it can be simply proved that the $AC_S(k)$ of the sequence S , for any k value, may be rewritten in terms of the CC 's or AC 's of the sub-sequences. As an example, $AC_S(2)$ is computed below, for a sequence S composed by three sub-sequences ($M = 3$) of length $l_s = 4$:

$$\begin{array}{cccccccccccc} a_{11} & a_{21} & a_{31} & a_{12} & a_{22} & a_{32} & a_{13} & a_{23} & a_{33} & a_{14} & a_{24} & a_{34} \\ a_{24} & a_{34} & a_{11} & a_{21} & a_{31} & a_{12} & a_{22} & a_{32} & a_{13} & a_{23} & a_{33} & a_{14} \\ \hline a_{11}a_{24} + a_{21}a_{34} + a_{31}a_{11} + a_{12}a_{21} + a_{22}a_{31} + a_{32}a_{12} + \\ a_{13}a_{22} + a_{23}a_{32} + a_{33}a_{13} + a_{14}a_{23} + a_{24}a_{33} + a_{34}a_{14} \end{array}$$

$$\Rightarrow AC_S(2) = CC_{A_1 A_2}(1, A_2^R) + CC_{A_2 A_3}(1, A_3^R) + CC_{A_3 A_1}(0, A_1^R) \quad (8)$$

To expand and generalize Eq.(8), aiming at a final sequence S featuring a near-optimal AC (i.e. an AC profile approaching a two-valued curve), we need to find the answers to the following questions. To simply understand the question, consider $CC_{A_i A_{\lambda}}(k, A_{\lambda}^R)$ as a model, based on which each term of Eq.(8) follows:

- 1) Which sub-sequences should be correlated together? (Finding i and λ)

TABLE III
SUBSCRIPTIONS OF THE ELEMENT OF THE SEQUENCE S FOR THE M -TUPLE

k	(j, q)				
	(1,1)	(2,1)	(3,1)	...	(M, l_s)
1	(M, l_s)	(1,1)	(2,1)	...	(M-1,1)
2	(M-1, l_s)	(M, l_s)	(1,1)	...	(M-2,1)
...
M	(1, l_s)	(2, l_s)	(3, l_s)	...	(M, l_s)
M+1	(M, l_s -1)	(1, l_s)	(2, l_s)	...	(M-1, l_s)
M+2	(M-1, l_s -1)	(M, l_s -1)	(1, l_s)	...	(M-2, l_s)
...
2M	(1, l_s -1)	(2, l_s -1)	(3, l_s -1)	...	(M, l_s -1)
2M+1	(M, l_s -2)	(1, l_s -1)	(2, l_s -1)	...	(M-1, l_s -1)
2M+2	(M-1, l_s -2)	(M, l_s -2)	(1, l_s -1)	...	(M-2, l_s -1)
...
M l_s	(1,1)	(2,1)	(3,1)	...	(M,1)

- 2) How many times the corresponding sub-sequence should be shifted? (Finding k)

As it is clear from the terms in Eq.(8), and due to the periodic repetition of all the sub-sequences, the first sub-sequence at each correlation term (subscript i) is fixed for any k value, and i goes from 1 to M . The second sub-sequence at each correlation term (subscript λ) depends on the k value and the sequence number. All the parameters in the correlation terms can be found by the subscriptions of the elements of the sequence S . Table III shows the subscriptions of the first M -tuple of the sequence S : the first argument in each parenthesis identifies the sub-sequence number, and the second argument is the bit position in that sub-sequence.

Since the sub-sequences are periodically repeated, as mentioned above, all the sub-sequence numbers may be found at any M -tuple in the first argument. Likewise, the first argument in parenthesis in the first column periodically repeats $[M, M-1, \dots, 2, 1]$. In fact, we need to map the shift index k to the mentioned matrix. Therefore, Eq.(9) may be used for mapping, where j and λ are the sub-sequence numbers of the non-shifted (shown by blue color in Table III) and shifted sequence (shown by red color in Table III), respectively. Moreover, $Seq_Ind = [1, 2, \dots, M-1, M, 1, 2, \dots, M-1, M]$.

$$\lambda = Seq_Ind(M - (k - 1) + (j - 1)), j = 1 \dots M \quad (9)$$

It is important to note that Eq.(9) does not work for $k > M$, as k continuously increases from 1 to $M \cdot l_s$ and $[M, M-1, \dots, 2, 1]$ is periodically repeated in the first argument of the first column. Thus, the k factor should be modified in Eq.(9). Eq.(10) may be employed to tackle this problem. In fact, parameter k in Eq.(9) should be substituted by the x of Eq.(10). In Eq.(10), mod is the modulo function.

$$x = T(\text{mod}(k, M) + 1), \quad T = [M, 1, 2, \dots, M-1] \quad (10)$$

The same way is utilized to formulate the shift index mentioned in Eq.(7). The following table shows the amount of shift for each sub-sequence at each rotation. Table IV is derived from Table III. This table may be formulated according

to Eq.(11), where x is obtained from Eq.(10) and $\lfloor \cdot \rfloor$ is the floor function.

$$Shf_{in}(k) = \begin{cases} \lfloor \frac{k}{M} + 1 \rfloor & \text{if } x \geq i \\ \lfloor \frac{k}{M} \rfloor & \text{if } x < i \end{cases} \quad (11)$$

TABLE IV
SHIFT INDEX VALUE AT EACH ROTATION

k	1	2	3	...	M
1	1	0	0	...	0
2	1	1	0	...	0
...
M	1	1	1	...	1
$M+1$	2	1	1	...	1
$M+2$	2	2	1	...	1
...
$2M$	2	2	2	...	2
$2M+1$	3	2	2	...	2
$2M+2$	3	3	2	...	2
...
$M \cdot l_s$	0	0	0	...	0

IV. OPTIMIZATION OF THE GENERATION PROCESS

A. Background

PSO is a metaheuristic optimization approach inspired by bird flock simulation. The original PSO (OPSO) is proposed by Kennedy and Eberhart [20]. Since OPSO is unstable, many works have been presented to improve the algorithm from the divergence point of view. Several papers in the literature, like [35], [36], show that PSO has been applied to solve problems with continuous, discrete, and even mixed variables.

In OPSO, *particles* are the sets of variables which form a potential solution. The set of particles is called *swarm*. The particles *fly* in a multidimensional space to find the optimal solution. Each particle has a position and a velocity. According to the PSO algorithm, different factors determine the new velocity of each particle. The velocity in the previous position is one of the parameters which take effect on the new velocity value. Likewise, two distances are used to find the new velocity. The former is the distance between the current position of each particle, and the position in which it had its own best experience so far. The latter is the distance between the current position of each particle, and the best position with respect to all the positions that all the particles experienced so far. After updating the velocity of each particle, their position will be updated by the new velocity and the position taken in the last iteration. Considering the explained strategy, the OPSO is formulated as per Eq.(12):

$$\begin{aligned} v_{ij}^t &= v_{ij}^{t-1} + c_1 r_1 (Pbest_{ij}^{t-1} - x_{ij}^{t-1}) + \\ & c_2 r_2 (nbest_{ij}^{t-1} - x_{ij}^{t-1}) \\ x_{ij}^t &= x_{ij}^{t-1} + v_{ij}^t \end{aligned} \quad (12)$$

where the parameters have the following definitions:

v_{ij}^t, x_{ij}^t : the component (velocity and position) j of particle i

at time/iteration t ;

$Pbest_{ij}^t, nbest_j^t$: the best experience of each particle and swarm, respectively, at time/iteration t ;

c_1, c_2 : individual and social acceleration constants;

r_1, r_2 : uniform distributed random values.

A *fitness function*, $f(\cdot)$, is a function of all the variables that shows the goodness of each particle. In fact, $Pbest$ and $nbest$ are selected from the swarms based on the value of $f(\cdot)$, for each particle. The selection of $nbest$ also depends on another factor which is the search type. In general, the search type may be global or local, respectively known as $gbest$ and $lbest$. The $gbest$ would perform if each known particle is able to communicate with all other particles. In other words, if the Fully Connected topology is selected as the neighborhood topology, the particles use the $gbest$ in the given space. Otherwise, the neighborhood topology implements the $lbest$. It is proved that PSO based on $lbest$ usually outperforms the $gbest$ [37]. Even if $gbest$ may have the faster convergence, $lbest$ has the benefits of better exploration and promising results [37], [38].

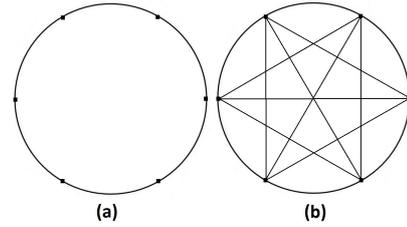


Fig. 1. (a) Ring and (b) Fully-Connected topology

Many researches have been carried out to evaluate different population topology options. Ring, Mesh, Star, Tree and so on are among the popular population topology models. Fully-Connected and Ring are the most popular utilized topology models, as illustrated in Fig. 1. In comparison to the Fully-Connected, the Ring topology has higher convergence rate [39].

As stated above, the OPSO is not stable, hereby, we need to use the suitable modified PSO. Different variations to Eq.(12) have been proposed to improve the OPSO. One popular way is the use of variable acceleration coefficients (c_1 and c_2). Another approach is to apply the so-called *inertia weight* and *constriction factor* [40], [41]. The simultaneous adoption of different variations is to be carefully evaluated, as some of the improvements are not compatible. Other works proposed to improve the PSO algorithm by applying the *mutation* [42], [43].

B. PSO applied to the proposed generation method

In this section, the described PSO algorithm is applied to the constructed sequences based on Section III. The DBSs are employed to generate longer sequences, that are named as New Shuffled DBSs (NSDBSs). The AC of the NSDBSs depends on the CC and AC of each sub-sequence, as stated in the previous section. To achieve an appropriate AC of the

new longer sequence, we need to find out the best starting bit position (SBP) of each sub-sequence involved in *NSDBSs*. The PSO algorithm is used to find the SBPs of the sub-sequences. Since DBSs of span 5 are used to generate the *NSDBSs*, the bit position changes between 1 and 32. Therefore, the integer PSO algorithm should be implemented.

As mentioned above, OPSO is unstable and it should be modified to improve its performance. The first technique relies on employing the inertia weight (ω), which is proposed by [44], to have a better convergent behavior. The effect of this parameter is on the speed of particles at their previous position, as shown in Eq.(13). The use of a time decreasing ω , from 1.2 to 0.9, in [44], is also recommended, although the time decreasing function is still an open issue. The second technique, which also deals with the algorithm convergence, relies on the constriction factor (*CFa*). This parameter was initially studied by Clerc and Kennedy [45].

$$\begin{aligned} v_{ij}^t &= CFa [\omega * v_{ij}^{t-1} + c_1 r_1 (Pbest_{ij}^{t-1} - x_{ij}^{t-1}) + \\ & c_2 r_2 (nbest_{ij}^{t-1} - x_{ij}^{t-1})] \\ x_{ij}^t &= x_{ij}^{t-1} + v_{ij}^t \end{aligned} \quad (13)$$

The two factors ω and *CFa* explained above are obtained from Eq.(14) and Eq.(15), respectively, when ω_{min} and ω_{max} are the minimum and the maximum inertia weight. As given in Eq.(15), ϕ (i.e. the sum of c_1 and c_2) should be greater than 4, hereby, $c_1 = c_2 = 2.05$ if the constriction factor is applied.

$$\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{Iteration_{max}} Iteration \quad (14)$$

$$CFa = \frac{2}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|}, \quad \phi = c_1 + c_2 > 4 \quad (15)$$

The use of variable acceleration coefficients (VAC) (c_1 and c_2) may also be applied to have better performance in comparison with OPSO. This strategy is proposed in [46]. In fact, two concerns are mentioned in [20] related to assigning large values to c_1 and c_2 . First, having large value of c_1 leads to clustering the particles around the local optimal solution. Second, considering a large value for c_2 causes an unpleasant effect which is overlooking the search space. Thus, VAC is employed to cope with these issues and then we propose a modification to have a superior result in shorter time. In VAC, both c_1 and c_2 are decreasing and increasing linearly, respectively, over the simulation time. Eq.(16) indicates the general formula for both of the acceleration coefficients where c_i and c_f are the initial and the final corresponding acceleration coefficients [46]. In this case, c_1 and c_2 change from 3 to 0.5 and from 0.5 to 3, respectively. Actually, c_i and c_f depend on the problem, however, the general trend is the same as shown in Fig. 2.

$$c = (c_f - c_i) \frac{Iteration}{Iteration_{max}} + c_i \quad (16)$$

Considering the roles and effects of c_1 and c_2 , it can be argued that changing the trend of acceleration coefficients could be helpful to optimize the results. As a matter of fact,

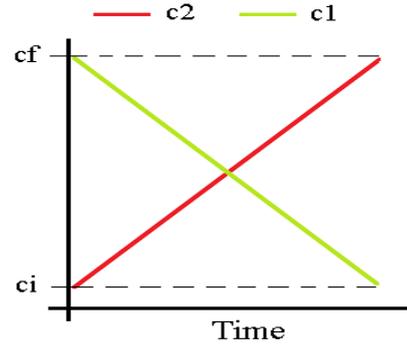


Fig. 2. The trend of acceleration coefficients over the time

there are two states possible in this method: c_1 and c_2 change between $[0.5, 2.5]$ and $[2.5, 0.5]$, and vice versa.

Variable population size (VPS) can be stated as an approach to achieve lower fitness function and convergence time. The dynamic population size for PSO was first introduced by Lei [47]. Two ways of diminishing and expanding PSO population size (DP-PSO, EP-PSO) are presented in [47], to implement the VPS improvement of OPSO. In EP-PSO, the more particles are covered, in contrast with DP-PSO. Note that in DP-PSO, the particles with the largest fitness function are removed from the swarm. It is preferred to have global and local optimum search in EP-PSO and DP-PSO, respectively. Moreover, it is attempted to include both EP-PSO and DP-PSO in the simulation process. The only reason is that it helps to avoid being trapped in either local or global optima. Therefore, there are two states during the simulation: the initial state is DP-PSO. If the simulation runs for 100 iterations, and the fitness function of *nbest* dose not change, the state would switch to another one. The number of iteration set to change the state depends on the nature of the problem.

V. SIMULATIONS AND RESULTS

The Root-Mean-Square Error (RMSE) function is adopted to calculate the fitness function. It is due to the fact that the ideal AC side-lobes are equal to zero, thereby, it is desirable to achieve the set of sequences with zero out-phase AC. Thus, the RMS of the difference between the AC side-lobes of the sequences output from the simulations, and the ideal one, may be an appropriate function to calculate the error. The following configurations are applied to all the simulations executed:

- The population size is set to 40.
- At the beginning, the position of each particle is initialized by an integer random number generator.
- The initial velocity of each particle is zero.
- The maximum number of iterations is 1000.
- The aim is to minimize the fitness function $f(\psi) = RMSE(AC(k)); 1 \leq k \leq N$, where N is the number of side-lobes in the AC function.
- Both the Fully-Connected and Ring topology are implemented.

- The number of DB sub-sequences used to generate a sequence of length 2048 bit is 64.

In Table V, the number of iterations at which the fitness function reaches its minimum value (before the 1000 admitted iterations), for each simulation, is indicated. If a technique is applied to the algorithm it is indicated by "1", otherwise "0". In the fourth column, *Type1* and *Type2* refer to the VAC technique itself and its modification, respectively. The provided results in Table V have been averaged over 10 simulation runs. Regarding to the programming aspect, PSO and all the mentioned approaches used to improve the OPSO, have been implemented in MATLAB language.

TABLE V
AVERAGE AMOUNT OF ITERATIONS NEEDED TO GET THE OPTIMAL RESULTS, FOR DIFFERENT OPSO TECHNIQUES APPLIED TO NSDBS GENERATION

Test #	VPS	Variable ω	CFa	VAC		# Iterations	
				(Type1/Type2)	Fully-Connected	Ring	
1	0	0	0	1-Type1	864.1	588.7	
2	0	0	0	1-Type2	294.2	328.5	
3	0	0	1	0	285.8	715.9	
4	0	1	0	1-Type1	943.4	566.9	
5	0	1	0	1-Type2	869.4	474	
6	0	1	1	0	842.4	728.3	
7	1	0	0	1-Type1	954.4	703.8	
8	1	0	0	1-Type2	336.7	552.5	
9	1	0	1	0	504	650.1	
10	1	1	0	1-Type1	762.6	812.1	
11	1	1	0	1-Type2	853.6	840.2	
12	1	1	1	0	912.1	781.2	

Considering the fact mentioned in Section III-B, about the possibility of suppressing the AC effect of each sub-sequence, the selected sub-sequences may have a great impact on the results. The important point to focus on is that all the tests are executed over the same set of sub-sequences, to gather meaningful results for comparison. It means that the best results may also be obtained if the sub-sequences are carefully selected.

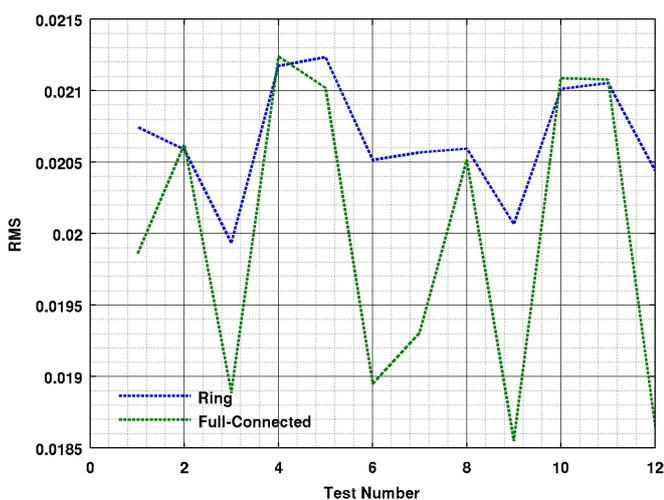


Fig. 3. Comparison of *rms* function of AC side-lobes of two topology of Ring and Full-Connected for each tests

Fig. 3 compares the minimum value of the fitness function for the Fully-Connected and Ring topology. In the figure, the horizontal axis has the same order of Table V. As evident from the results in Fig. 3, the Fully-Connected topology outperforms the Ring topology, specifically when neither the VAC

technique nor its modification is applied (see tests numbered 3, 6, 9 and 12). In the test number 9, PSO finds the optimal result among the other tests. This test shows the impact of VPS and *CFa* on the PSO algorithm.

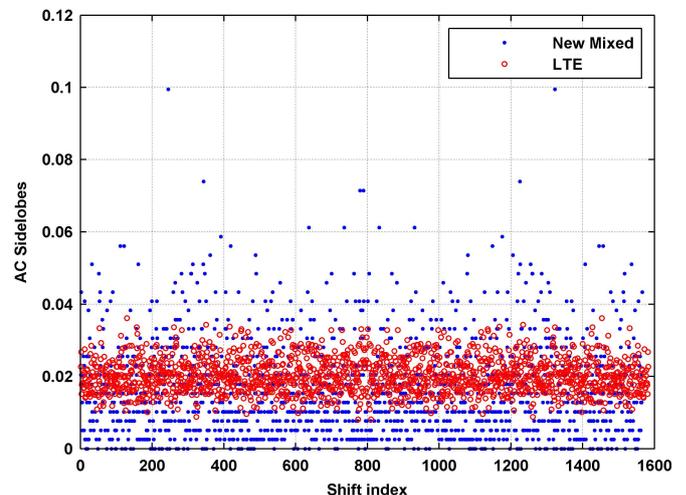


Fig. 4. Comparison of LTE and NSDB sequences

The test 9 is also performed on the DBSs to generate sequences of length 1568 bits. This specific case is motivated by the intention of comparing the resulting sequences to the ones generated and used in Long Term Evolution (LTE) systems, as provided also within the LTE simulator by Vienna University [48]. According to the simulation results, the average RMS of AC side-lobes of the NSDBS and the sequences used in the LTE simulator are 0.021 and 0.2054, respectively, which motivates the slightly superior results of NSDBS over the sequences used in LTE. Fig. 4 shows clearly the higher performance of NSDBS. It is also notable that, even if there is not significant difference between the RMS function values, unlike the LTE sequences, many zero AC side-lobe may be found in NSDBS.

It can be seen clearly from Fig. 4 that the maximum side-lobe of NSDBSs occurred at a shift index value approximately equal to 0.1. The shift indexes of the maximum side-lobe are 245 and 1323. According to Eq.(7), the AC of NSDBSs at these shift indexes is equal to the sum of AC of sub-sequences at 5^{th} and 27^{th} shift indexes. Interestingly, these are related to the ZCZ of DBSs. In fact, a DBS of span 5 has a ZCZ of length 4; therefore 5^{th} and 27^{th} shift indexes are exactly next to the ZCZ. Likewise, the AC value of DBSs at these shift indexes (5^{th} and 27^{th}) has different values, which are -0.6250 , -0.3750 , -0.1250 and 0.1250 in the case of DBSs of span 5. As a result, the maximum side-lobe of NSDBSs can be substantially degraded if the sub-sequences are appropriately selected such that their AC values mutually cancel each other. The maximum side-lobe may be even set to zero if an even number of sub-sequences are used to generate the resulting optimal sequences.

Run-time complexity is another important parameter that can be considered to assess the algorithm performance. As mentioned in Section II, the work by Mow et al. [27], to the best of our knowledge is the only paper that intended to

generate binary sequences with low AC by the use of EA. As reported in [27], the generation of binary sequence of length 1019 required 16.1136 hours, and the time needed exhibited a quadratic growth with the sequence length. The simulation results performed by us indicate that we are able to generate the appropriate set of binary sequences in 19 minutes. Likewise, the implemented code is run for other different lengths of 512, 1024 and 2048 to show that the proposed technique works appropriately. The simulation approximately took 3.5, 9.5 and 31 minutes, respectively. It may be stated that the run-time would be increases 3 times if the requested sequence length is doubled. Consequently, our proposed technique provides superior run-time performance, for the same RMS value of the AC side-lobe, approximately.

VI. CONCLUSION

In this paper, a method to generate long sequences, generalized to the use of M sub-sequences, has been introduced, aimed at minimizing the AC side-lobes of the resulting sequences, by exploiting the AC properties of the sub-sequences. According to the results, the AC side-lobes of the longer sequences are obtained as the combination of AC and CC values of the sub-sequences. Thereby, it is possible to control the desired AC profile of the new sequences, so that it approaches the ideal Dirac delta function. To find the best starting bit position for each sub-sequence, the Particle Swarm Optimization algorithm and its improvements have been applied.

The paper showed that mixing De Bruijn sequences according to the aforementioned approach, provides remarkable performance in terms of suppressing the AC side-lobe values of the longer sequence. All the modified PSO versions have been applied on the Fully-Connected and Ring topology. Taking into account the comparison of the achieved outcomes, we may state that the use of the Fully-Connected topology outperforms the Ring topology. As a possible application case study, it has been shown that the new sequences, resulting from the aforementioned construction method, joint the PSO algorithm, may be also utilized in LTE, thanks to their lower AC side-lobes values.

Likewise, the simulation time has been estimated, to evaluate the complexity of the process. The comparison between the obtained results and those provided in [27] shows that the proposed sequence construction methodology may generate the longer binary sequence in substantially shorter time. Note that, in order to limit the complexity of the proposed construction method, the order of the sub-sequences is fixed in the PSO algorithm, and the starting bit position is the only parameter determined by PSO. It means that the algorithm may result in sequences with even lower AC side-lobes if the order of the sub-sequences is taken into account as another parameter for optimization.

REFERENCES

- [1] S. Ilcev, "Implementation of multiple access techniques applicable for maritime satellite communications," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 7, no. 4, 2013.
- [2] J. D. Taylor, *Ultra-wideband radar technology*. CRC press, 2000.
- [3] V. P. Ipatov and S. B. G. CdMA, "Some proposals on signal formats for future gnss air interface," *Inside GNSS*, vol. 5, no. 5, pp. 46–51, 2010.
- [4] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS GPS, GLONAS, GALILEO and more*. Springer-Verlag, 2008.
- [5] P. Sahu and A. Chaturvedi, "Application of multicarrier cdma to mobile communication technology," in *Industrial Technology 2000. Proceedings of IEEE International Conference on*, vol. 1, Jan 2000, pp. 427–431 vol.2.
- [6] H. Holma and A. Toskala, *WCDMA for UMTS. Radio Access for Third Generation Mobile Communications*, 2004.
- [7] B. G. Lee and B.-H. Kim, *Scrambling techniques for CDMA communications*. Springer Science & Business Media, 2001, vol. 624.
- [8] W. Sun, A. Klapper, and Y. X. Yang, "On correlations of a family of generalized geometric sequences," *IEEE Transactions on Information Theory*, vol. 47, no. 3, pp. 2609–2618, 2001.
- [9] Y. C. Liu, C. W. Chen, and Y. T. Su, "New constructions of zero-correlation zone sequences," *IEEE Transactions on Information Theory*, vol. 59, no. 8, pp. 4994–5007, 2013.
- [10] S. Hara and R. Prasad, "Design and performance of multicarrier cdma system in frequency-selective rayleigh fading channels," *Vehicular Technology, IEEE Transactions on*, vol. 48, no. 5, pp. 1584–1595, Sep 1999.
- [11] H.-H. Chen, *The Next Generation CDMA Technologies*. John Wiley & Sons, 2007. [Online]. Available: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:No+Title\#0>
- [12] S. Stanczak and H. Boche, "Influence of periodic correlation properties of sequences on the sum capacity of cdma systems," in *Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on*, vol. 2, Oct 2000, pp. 1288–1292 vol.2.
- [13] J. Singh, "Capacity enhancement of wavelength/time/space asynchronous optical {CDMA} with relaxed cross-correlation," *Optical Fiber Technology*, vol. 19, no. 6, Part A, pp. 614 – 617, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1068520013001259>
- [14] B.-H. Lee, Byeong Gi and Kim, *SCRAMBLING TECHNIQUES FOR CDMA COMMUNICATIONS*. Springer London, 2002.
- [15] R. Lloyd-Evans, *QoS in integrated 3G networks*. Artech House, 2002.
- [16] H. Kaaranen, A. Ahtiainen, L. Laitinen, S. Naghian, and V. Niemi, *UMTS Networks: Architecture, Mobility and Services: Second Edition*. John Wiley & Sons, 2005.
- [17] S. Spinsante and E. Gambi, "De bruijn binary sequences and spread spectrum applications: A marriage possible?" *Aerospace and Electronic Systems Magazine, IEEE*, vol. 28, no. 11, pp. 28–39, Nov 2013.
- [18] M. Sarayloo, E. Gambi, and S. Spinsante, "De bruijn spreading sequences for dense cdma-based wsns," in *Networks and Communications (EuCNC), 2014 European Conference on*, June 2014, pp. 1–5.
- [19] R. Hassan, B. Cohanin, O. De Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *Proceedings of the 1st AIAA multidisciplinary design optimization specialist conference*, 2005, pp. 1–13.
- [20] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.
- [21] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43 – 53, 2005. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1474034605000091>
- [22] L. Nian-sheng, "Pseudo-randomness and complexity of binary sequences generated by the chaotic system," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 2, pp. 761 – 768, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1007570410002091>
- [23] S. Babu Sekar, S. Sengupta, and K. Bandyopadhyay, "Zero correlation zone sequences for satellite navigation," in *India Conference (INDICON), 2011 Annual IEEE*, Dec 2011, pp. 1–4.
- [24] R. Mita, G. Palumbo, S. Pennisi, and M. Poli, "Pseudorandom bit generator based on dynamic linear feedback topology," *Electronics Letters*, vol. 38, no. 19, pp. 1097–1098, Sep 2002.
- [25] A. Peinado and A. Fster-Sabater, "Generation of pseudorandom binary sequences by means of linear feedback shift registers (lfsrs) with dynamic feedback," *Mathematical and Computer Modelling*, vol. 57, no. 1112, pp. 2596 – 2604, 2013, information System Security and Performance Modeling and Simulation for Future Mobile Networks. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0895717711004341>
- [26] M. Sonmez Turan, "Evolutionary construction of de bruijn sequences," in *Proceedings of the 4th ACM Workshop on Security and Artificial*

- Intelligence*, ser. AISec '11. New York, NY, USA: ACM, 2011, pp. 81–86. [Online]. Available: <http://doi.acm.org/10.1145/2046684.2046696>
- [27] W. H. Mow, K.-L. Du, and W. H. Wu, "New evolutionary search for long low autocorrelation binary sequences," *Aerospace and Electronic Systems, IEEE Transactions on*, vol. 51, no. 1, pp. 290–303, January 2015.
- [28] A. H. Chan, R. A. Games, and E. L. Key, "On the complexities of de Bruijn sequences," *Journal of Combinatorial Theory, Series A*, vol. 33, no. 3, pp. 233–246, 1982. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0097316582900383>
- [29] F. J. Mowle, "Relations between pn cycles and stable feedback shift registers," *Electronic Computers, IEEE Transactions on*, vol. EC-15, no. 3, pp. 375–378, June 1966.
- [30] K. B. Magleby, "The synthesis of nonlinear feedback shift registers," Ph.D. dissertation, Department of Electrical Engineering, Stanford University, 1963.
- [31] E. Leach, "Regular sequences and frequency distributions," *Proceedings of the American Mathematical Society*, vol. 11, no. 4, pp. 566–574, 1960.
- [32] A. Lempel, "On a homomorphism of the de bruijn graph and its applications to the design of feedback shift registers," *Computers, IEEE Transactions on*, vol. C-19, no. 12, pp. 1204–1209, Dec 1970.
- [33] N. De Bruijn, "A combinatorial problem," *Proc. Nederlandse Akademie van Wetenschappen*, 1946.
- [34] Z. Zhaozhi and C. Wende, "Correlation properties of De Bruijn sequences," *Syst Sci Math Sci Acad Sinica*, vol. 2, pp. 170–183, 1989.
- [35] H. Izakian, B. T. Ladani, A. Abraham, and V. Snasel, "A discrete particle swarm optimization approach for grid job scheduling," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 9, pp. 1–15, 2010.
- [36] H. Nahvi and I. Mohagheghian, "A particle swarm optimization algorithm for mixed variable nonlinear problems," *International Journal of Engineering*, vol. 24, no. 1, 2011.
- [37] A. Engelbrecht, "Particle swarm optimization: Global best or local best?" in *Computational Intelligence and 11th Brazilian Congress on Computational Intelligence (BRICS-CCI CBIC), 2013 BRICS Congress on*, Sept 2013, pp. 124–135.
- [38] R. Eberhart, P. Simpson, and R. Dobbins, *Computational intelligence PC tools*. Academic Press Professional, Inc., 1996.
- [39] G. Toscano-Pulido, A. J. Reyes-Medina, and J. G. Ramirez-Torres, "A statistical study of the effects of neighborhood topologies in particle swarm optimization," in *Computational Intelligence*. Springer, 2011, pp. 179–192.
- [40] R. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on*, vol. 1, 2000, pp. 84–88 vol.1.
- [41] M. S. Innocente and J. Sienz, "Particle swarm optimization with inertia weight and constriction factor," in *Proceedings of the International conference on swarm intelligence (ICSI11)*, 2011.
- [42] Y. Zhang, S. Wang, P. Phillips, and G. Ji, "Binary pso with mutation operator for feature selection using decision tree applied to spam detection," *Knowledge-Based Systems*, vol. 64, pp. 22–31, 2014.
- [43] P. Chou and J. Chen, "Enforced mutation to enhancing the capability of particle swarm optimization algorithms," in *Proceedings of the Second International Conference on Advances in Swarm Intelligence - Volume Part I, ser. ICSI'11*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 28–37. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2026282.2026287>
- [44] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on*, May 1998, pp. 69–73.
- [45] M. Clerc and J. Kennedy, "The particle swarm - explosion, stability, and convergence in a multidimensional complex space," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 1, pp. 58–73, Feb 2002.
- [46] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *Evolutionary Computation, IEEE Transactions on*, vol. 8, no. 3, pp. 240–255, June 2004.
- [47] C. Lei, "The study on dynamic population size improvements for classical particle swarm optimization," in *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol. 1, Dec 2011, pp. 430–433.
- [48] "TC: Vienna LTE-A Simulators," 2015. [Online]. Available: <http://www.nt.tuwien.ac.at/research/mobile-communications/vienna-lte-a-simulators/>



Mahdiy Sarayloo received his M.Sc. in 2011 from the University of Southampton (UK) for his research on System On Chip (SOC). Since November 2013 he is pursuing his Ph.D. at Università Politecnica delle Marche (Ancona, Italy) by working on the analysis and evaluation of the performance of De Bruijn sequences in multiuser communications, such as CDMA-based systems, and LTE.



Ennio Gambi was born in Loreto (Ancona, Italy) in 1961. He received the Electronic Engineering graduate diploma at the Università Politecnica delle Marche in 1986 and the Microwave Engineering master degree in 1989. From 1984 to 1992 he worked with the Azienda di Stato per i Servizi Telefonici, while during 1988 he covered the role of Avionic Engineer as an Official of the Italian Military Air Force. Since 1992 he joined the Università Politecnica delle Marche in Ancona, where he is, currently, an Associate Professor. At present

his main research interests are in spread spectrum systems, both for communication and radar purposes. In particular, his research is dedicated to the selection of suitable sets of expansion sequences, able to increase the system performance. At the same time, on the basis of a long time interest on source coding, he is working on the evolution of this technique in the field of Ambient Assisted Living, with the aim to propose an integrated solution where the fusion of data provided by different sensor networks allows to offer the desired level of home assistance for disabled or elderly people. He is coauthor of several international papers.



Susanna Spinsante received her Ph.D. in Electronic Engineering and Telecommunications in 2005, from the Polytechnic University of Marche (ITALY), and her Laurea (Graduate Diploma) Degree in Electronic Engineering in 2002, from the same University. Since December 2005, she has been a PostDoc in Telecommunications, at the same University, where now she is Assistant Professor in Telecommunications. At present, Susanna works on radio communication systems exploiting spread spectrum techniques, for indoor and outdoor communications, and

radar applications. She is IEEE Senior Member, member of AitAAL (Italian Association for Ambient Assisted Living) since February 2010, and member of AEIT.