Journal of Computer Science

Original Research Paper

Simulated Annealing Decoder for Linear Block Codes

[1]Lahcen Niharmine, [2]Hicham Bouzkraoui, [3]Ahmed Azouaoui and [4]Youssef Hadi

[1]*SIME Lab, ENSIAS, Mohamed V University, Rabat, Morocco*
[2,4]*MISC Laboratory, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco*
[3]*Department of Computer Science, Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco*

Article history
Received: 24-06-2018
Revised: 13-08-2018
Accepted: 1-09-2018

Corresponding Author:
Lahcen Niharmine
SIME Lab, ENSIAS, Mohamed V University, Rabat, Morocco
Email: niharmine@gmail.com

Abstract: In this study, we introduce a novel soft decoder, the first of its kind, for linear block codes, based on Simulated Annealing algorithm (SA). The main enhancement in our contribution which let our decoder over performs with large gain (about 3 dB at 7×10-4) the classical SA approach, is to take the most reliable information set of the received codeword as a start solution and also according to this reliability generate neighbor's solutions. Besides, our algorithm performance is enhanced by reducing search space when we involve the code error correcting capability parameter. The performance of the designed algorithm is investigated through a parameter tuning process and then compared with other various decoding algorithms in terms of decoding performance and algorithmic complexity. Simulation results, show that our algorithm over performs its competitor decoders while keeping minimum computation cost. In fact, our algorithm has large gain over Chase-2 and GAMD, furthermore, it over performs the most efficient and up to date DDGA decoder by 2 dB at 10-5 for RS codes.

Keywords: Error Correcting Codes, Simulated Annealing, Soft Decoding, Linear Codes, Metaheuristic

Introduction

After the emergence of error correcting codes theory, several researches were proposed to design "good" and practical decoders. The goal of these decoders is to minimize the probability of decoding error. Ideally, the Maximum Likelihood Decoding (MLD) is the best approach, but the decoding process is impracticable for large codes because all code words need to be analyzed. Scientists have designed techniques that convert this problem into graph optimization. These algorithms become impracticable or inefficient for large codes, although the use of long code is important in communication design to approach channel limit as proven by Shannon's landmark paper (Shannon, 1948). When metaheuristic and probabilistic algorithms became widely used and recognized as efficient approaches for hard optimization problems, in the first instance, hard decoding was coupled with these techniques to design good decoders by making hard decision on the received signal then trying to find the most probable transmitted code using metaheuristic and probabilistic search. Prange (1962) proposed a decoding algorithm using information set, this method was enhanced and used later by several researchers (Lee and Brickell, 1988), (Leon, 1988) and (Stern, 1989). Recently, Genetic Algorithms are widely introduced in this field, actually Cardoso and Arantes (1999) have proposed a hard decoder based on GA, also several papers have been published in the same context (De Bona and Junior, 2015) and (Hoffman *et al*., 2011). It is also convenient to cite the work of Aylaj and Belkasmi (2015) by proposing a hard decoder based on SA method. However, such model discards all information about the reliability of the received signal and therefore losing accuracy in decoding capability.

In this paradigm, soft decoding decision was the alternative to take benefit from the information reliability directly from demodulation (Fig. 1), it uses the real values and decision uses euclidean distance, hence corrects more errors and reaches high performance, actually it can provide 2dB gain compared with the hard decoding (Proakis, 1995).

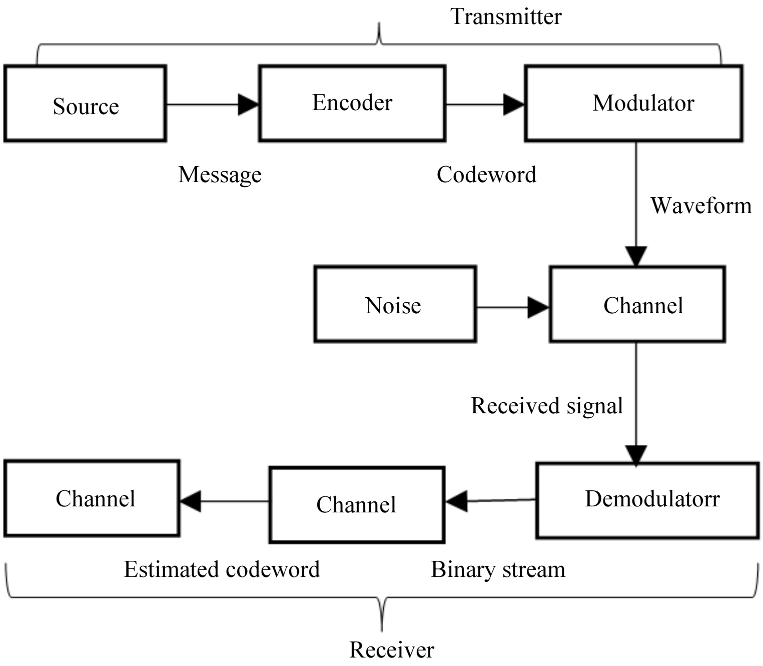

Fig. 1: Communication system model

Soft-decoding is considered as a NP-hard problem (Berlekamp *et al*., 1978), in this case classical exhaustive search would not be practicable anymore in large space. In this context, several algorithms were proposed to overcome this handicap. Forney (1966) proposed a pseudo soft decoding system, called "Generalized Minimum Distance Decoding" GMDD. Chase (1972) proposed a soft decoder based on channel measurement information. After the emergence of Artificial Intelligence and metaheuristic methods, several works were published of decoders based on neural networks and Genetic Algorithms (GA). In fact GA seems to give good results which is proved by Maini *et al*. (1994), then several GA came after such as the SDGA (Azouaoui and Belkasmi, 2012), the AutDAG decoder by (Nouh *et al*., 2013) and Shakeel (2010) GA-Based Decision Soft decoder. In contrast of the most decoders, other algorithms were proposed using dual code, such as CGAD (Azouaoui *et al*., 2012a), then recently two algorithms, CGAD-M and CGAD-HSP were proposed based on CGAD (Berkani *et al*., 2017), also Maini was enhanced to DDGA decoder (Azouaoui *et al*., 2012b), the SDGA was also extended to DSGA (Azouaoui *et al*. 2012c). In other hand for LDPC codes GAMD decoder (Scandurra *et al*., 2006) was proposed. In this spirit, we propose in this paper, a novel soft decoder based on Simulated Annealing Algorithm (SASD).

Simulated Annealing is a physical heating process used in metallurgy. Actually physical systems reach rapidly the equilibrium state (minimum energy) by decreasing slowly the temperature, this behavior is exploited to design an optimization algorithm. In this paper we will apply this technique on linear codes. The remainder of this article will be organized as follows: In section 2, we will introduce the Simulated Annealing Algorithm (SA), then in section 3 we will define the soft decision decoding problem context, then we will explain in details our proposed SA algorithm. In the next section we will investigate the simulation results firstly, by running parameter tuning process to find the best configuration for our decoder, afterward we compare our decoder with the classical SA algorithm and later we will focus on the SASD performances compared to its concurrent decoders. Finally we investigate the influence of the error correcting capability on our algorithm performance. In the last section we present the algorithmic complexity of our algorithm and its competitor. After all, we conclude our paper by summary of this work and the future perspectives.

Materials and Methods

Simulated Annealing

Simulated Annealing algorithm is a metaheuristic method used in optimization problems. It is derived from the process of annealing in metallurgy. In this process a metal is heated and slowly cooled under specific conditions to increase the size of the crystals in the metal and reduce their defects. The heat increases the energy of the atoms while the slow cooling allows to reach equilibrium state with a low energy. Each state of metal represents a different energy level of the system mostly modeled by Boltzmann distribution (Aarts and Korst, 1989). Heating the system results in exploring different states of the metal. When the system is cooled, the acceptance criteria will lead the changes to convergence. Once the system has cooled, the configuration will represent a state at or close to a global optimum.

This natural process was exploited to design an optimization algorithm used in several research fields. (Kirkpatrick *et al*., 1983) introduced SA in combinatorial problems, then several researches were published based on this approach (Hajek, 1988). In coding theory, (El Gamal *et al*., 1987) were among the first who used SA to design good codes. SA was also used to tackle the famous problem of finding the minimum distance of error correcting codes by Muxiang and Fullong (1994) then later by Aylaj and Belkasmi (2014).

The SA algorithm is allowed during execution, with a probabilistic model, to accept solutions that are worse than the current one, this gives the ability to escape from local optimum and explore more the search space. The following pseudo code represents a simple implementation of Simulated Annealing:

```
Parameter Initialization
$N_i$: number of iterations
$T_s$: starting temperature
$T_f$: final temperature
$S_0$: start solution
T←$T_s$
$S$←$S_0$
While ($T > T_f$)
{
While (iteration <$N_i$)
    {
        Pick a random neighbor solution ($s_n$);
        Evaluate $dE = E(s_n) – E(s)$;
        If dE≤ 0 then $s$←$s_n$;
        else if random(0,1) ≤ Exp(-dE/T) then
        $s$←$s_n$;
        end if
        iteration←iteration+1;
    }
T←cool(T);
}
```

The convergence proof suggests that with long enough cooling period this algorithm converges to the global optimum. In other hand the performance could be improved by using a good neighbor selection strategy, which we will show later in this paper.

Soft Decision Decoding as an Optimisation Problem

let $F2$ the binary field and note $C(n,k,d)$ a linear code of length n, dimension k and minimum distance d, we note also t the error correcting capability of C. This code can be represented by a $k \times n$ matrix G over $F2$ called generator matrix, a message m can be then encoded as follows:

$$c = mG \tag{1}$$

In other side, we define a parity check $(n\text{-}k) \times n$ matrix noted H which satisfies $HG^T = 0$, then we define for every vector $x \in F_2^n$ a syndrome $S(x)$:

$$S(x) = xH^T \tag{2}$$

If the codeword x contains no error then the syndrome $S(x)$ is zero.

In our study, the source generates a message m which is encoded to a codeword $c = (c_1,c_2,\ldots c_n)$ using (1), then BPSK-modulated to a signal $u = (u_1,u_2,\ldots u_n)$ where:

$$u_i = 2c_i - 1 \tag{3}$$

This signal is sent over a Gaussian channel, perturbed by an AWG noise which is modeled by a random n-vector $n = (n_1,n_2,\ldots n_n)$, with iid components given by n_i~N (0, N0 /2). In the receiver side the received signal is $r = (r_1,r_2,\ldots r_n)$ such that $r = u + n$. The likelihood probability is given by:

$$f_{r/u} = \frac{1}{(\pi N_0)^{n/2}} \exp\left(\sum\nolimits_{i=1}^{n} \frac{-(r_i - u_i)^2}{N_0} \right) \tag{4}$$

Clearly MLD can be formally expressed as an optimization problem as follows:

Given a received word r, what is the codeword $c \in C$ which maximizes the likelihood probability $f_{r/u}$?

$$/c \in C\} \leftrightarrow \min\left\{ \sum\nolimits_{i=1}^{n} (r_i - u_i)^2 / c \in C \right\} \tag{5}$$

Consequently, this problem is reduced to finding the minimum euclidean distance to the received word r overall codewords $c \in C$, this optimization has n variables out of which only k form a generator base, hence we could restrict search space to k variables, in preference we select the most k independent reliable bits, this will initialize the search closer to the global optimum.

Proposed Algorithm

We conceptualize the Euclidean distance between the received word and a codeword as the energy level and the state as the k bits vector, then the lowest energy state corresponds to the closest code word to the received word. The mapping between physical Simulated Annealing and our algorithm is summarized in the Table 1.

Table 1: Mapping between physical Simulated Annealing and our algorithm

Physical SA	SASD
State	k-bit vector
Energy	Euclidean distance between the received word and a codeword
Final State	Decoded word

When receiving a word $r = (r_1, r_2, \ldots r_n)$ our algorithm starts processing as follows:

1. Make a hard decision $\bar{r} = (\bar{r}_1, \bar{r}_2, \ldots \bar{r}_n)$ of received signal $r = (r_1, r_2, \ldots r_n)$:

$$\bar{r}_i = \begin{cases} 0, & r_i < 0 \\ 1, & r_i \geq 0 \end{cases} \tag{6}$$

2. Calculate the syndrome $S(\bar{r}) = \bar{r}H^T$, if $S(\bar{r}) = 0$, then $\bar{r}$ is estimated as the sent codeword and stop, otherwise continue
3. Sort the sequences $r = (r_1, r_2, \ldots r_n)$ in decreasing order based on reliability ($|r_i| \geq |r_{i+1}|$) to obtain new sequences $r' = (r'_1, r'_2, \ldots r'_n)$, let's note π the permutation $r' = \pi(r)$, we apply π to G to obtain G'. The most k reliable $(r'_1, r'_2, \ldots r'_k)$, will be the start solution of our algorithm, in fact this solution is estimated to be very close to the optimum solution, because if the received signal contains errors it's probable to happen in the (n-k) least reliable bits
4. Apply Gaussian elimination to G' to obtain a systematic matrix
5. We apply the SASD algorithm to obtain the best close word to the received signal and in consequence the estimated sent codeword c'. This algorithm is detailed afterwards
6. The code c' is related to the G' matrix, thus our estimated transmitted codeword is then:

$$\hat{c} = \pi^{-1}(c') \tag{7}$$

The SASD algorithm which is used to estimate the codeword will start from an initial state represented by a vector $s \in F_2^k$ and uses the euclidean distance as objective function to evaluate the quality of the solution which corresponds to the energy of the system and as we know the SA algorithm tries to converge toward a state with minimum energy, which is equivalent in our space to the closest codeword to the received word. In other side, in contrast with classical SA which generates neighbor's solution by random bit flipping, our algorithm takes again advantage of information reliability of the start solution vector $\bar{r}' = (\bar{r}'_1, \bar{r}'_2, \ldots \bar{r}'_n)$ the hard decision of sequences $r' = (r'_1, r'_2, \ldots r'_n)$, to generate next solution as follows:

```
pick_neighbor ()
{
    For each bit $i = k$ to 1
        If $rand(0,1) > \frac{1}{1+\exp\left(-2\frac{r'_i}{N_0}\right)}$ then
        flip the bit $r'_k$
        end if
end for
}
```

In the end The SASD algorithm could be expressed in the following pseudo code:

```
***Parameter Initialization***
***Ni:*** number of iterations
***Ts***: starting *temperature*
***Tf***: final temperature
***a***: cooling ratio
***S₀***: start solution
T←***Ts***
***S←S₀***

**While** (*T >Tf)*
**{**
**While** (*iteration* <Ni*)*
    {
      *s_n←pick_neighbor();*
Evaluate dE = E(s_n) – E(s);
Err=EvaluteCorrectedError();
        **If** *err<t* **then** *break;*
        **If** dE≤ *0* **then** *s←s_n;*
**else if** *random(0,1) ≤ Exp(-dE/T)*
      ***then***
*s←s_n;*
      **end if**
      *iteration←iteration+1*
    }
*T←a\*T;*
**}**
```

For every $s = (s_1, s_2, \ldots s_k)$ let's note $c = (c_1, c_2, \ldots c_n)$ the code word of s using (1). The energy $E(s)$ of $s = (s_1, s_2, \ldots s_k)$ is expressed as follows:

$$E(s) = \sum_{i=1}^{n} \left(r'_i - (2c_i - 1)\right)^2 \tag{8}$$

Results and Discussion

SASD Parameter Tuning

In order to select the most suitable set of parameters $\{\alpha, N_i, T_s, T_f\}$ we may choose some probabilistic models for estimation (MacKay, 2003). However in our case we assume that our parameters are independent and for this reason, we run several simulations of performance which is the Bit Error (BER) expressed as function of Signal to Noise Ratio (SNR), with different SASD parameters values. For every parameter optimization we set the remaining to default values according to the Table 2.

Parameter N_i

The number of iterations setting is often problem specific and estimated based on experiments. The simulations from the above Fig. 2, shows that the best performance is almost achieved when $N_i = 250$.

Parameter α

From the Fig. 3 α = 0.95, is almost the best choice for cooling ratio. In practice, slow cooling mechanism allows low euclideancodewords to be discovered and exploited.

Parameter T_s

Based on simulation results of Fig. 4, the best value for initial temperature T_s is 0.2. This parameter could impact accuracy, many methods have been proposed to estimate T_s (Ben-Ameur, 2004).

Parameter T_f

Intuitively, freezing temperature is used in physics to achieve equilibrium, this finding is confirmed by our simulations in Fig. 5, where we observe that the best performance is reached for almost all SNR values at the slowest temperature $T_f = 0.001$.

Comparison of the Proposed Algorithm Versus Classical Version

The proposed algorithm, SASD, was compared against the classical SA. The simulations where made with default parameters outlined in Table 2. The performances are given in terms of BER (bit error rate) as a function of Signal to Noise Ratio Eb/N0 (SNR).

Table 2: Parameters setting

Parameter	Value
Default code	BCH(63,45,7)
Channel	AWGN
Modulation	BPSK
Minimum number of bit error	200
Minimum number of blocs	1000
N_i	250
T_s	0.2
T_f	0.001
α	0.95

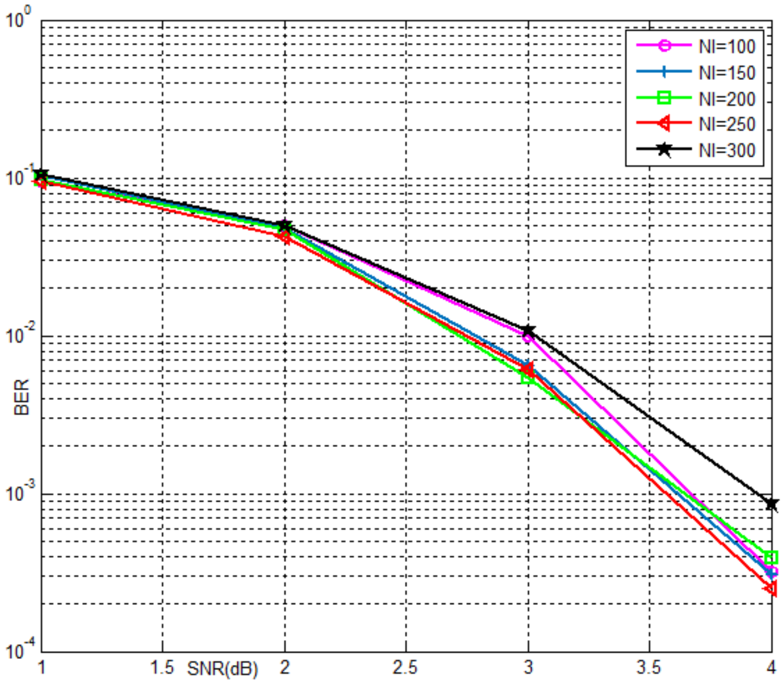

Fig. 2: Number of Iteration variation

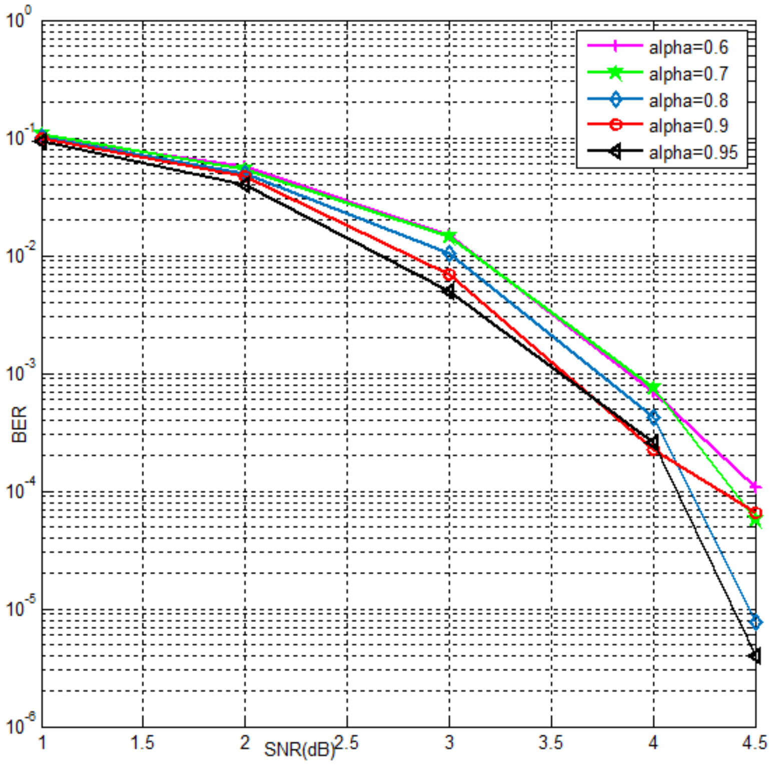

Fig. 3: Variation of parameter α

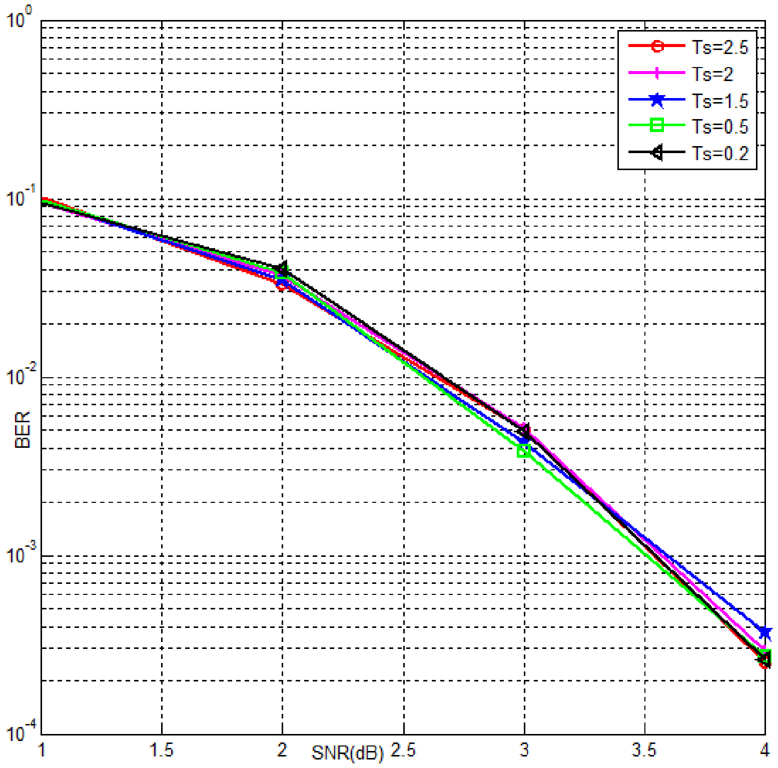

Fig. 4: Variation of parameter T_s

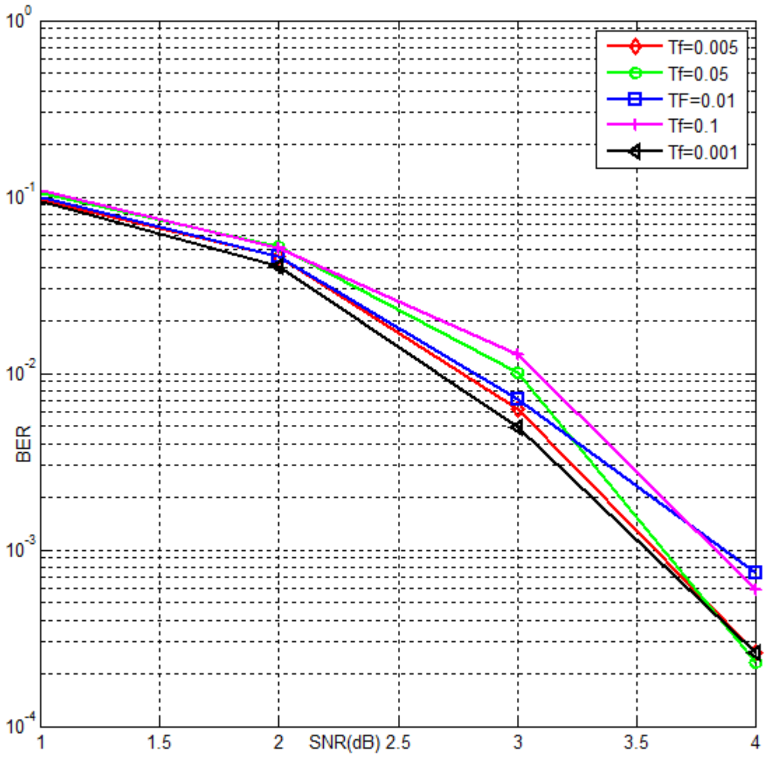

Fig. 5: Variation of parameter T_f

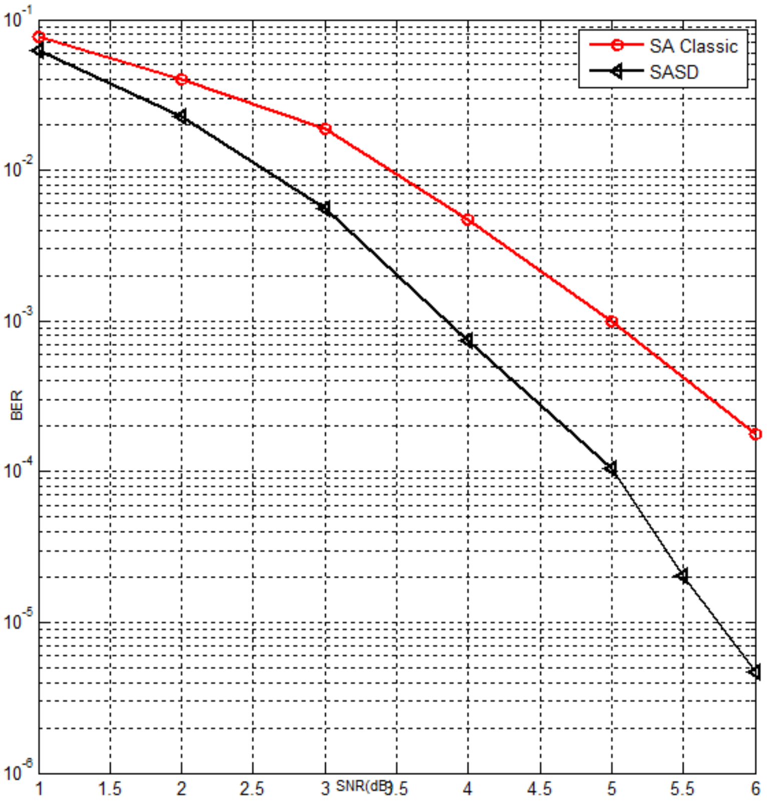

Fig. 6: Code BCH(31,21,5)

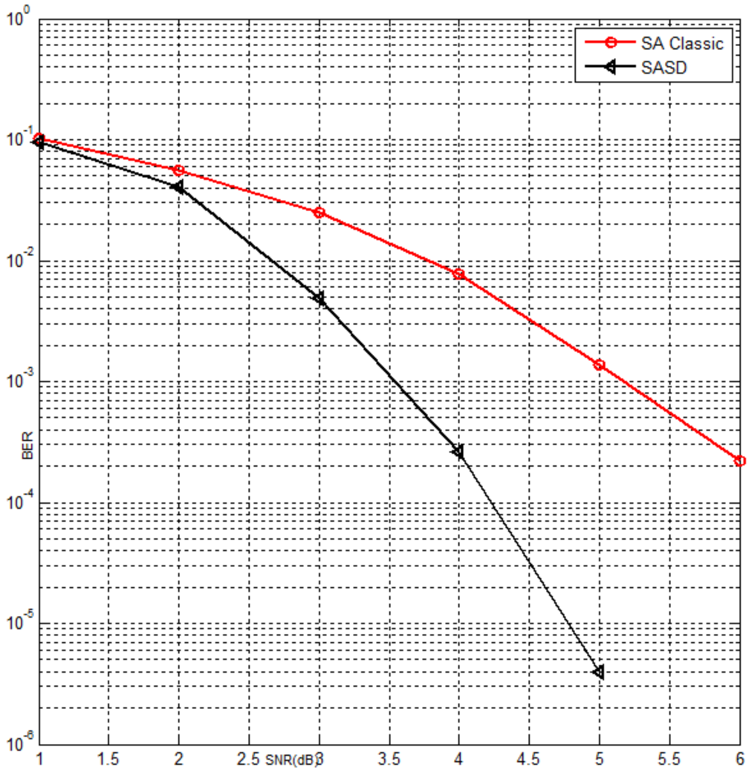

Fig. 7: Code BCH(63,45,7)

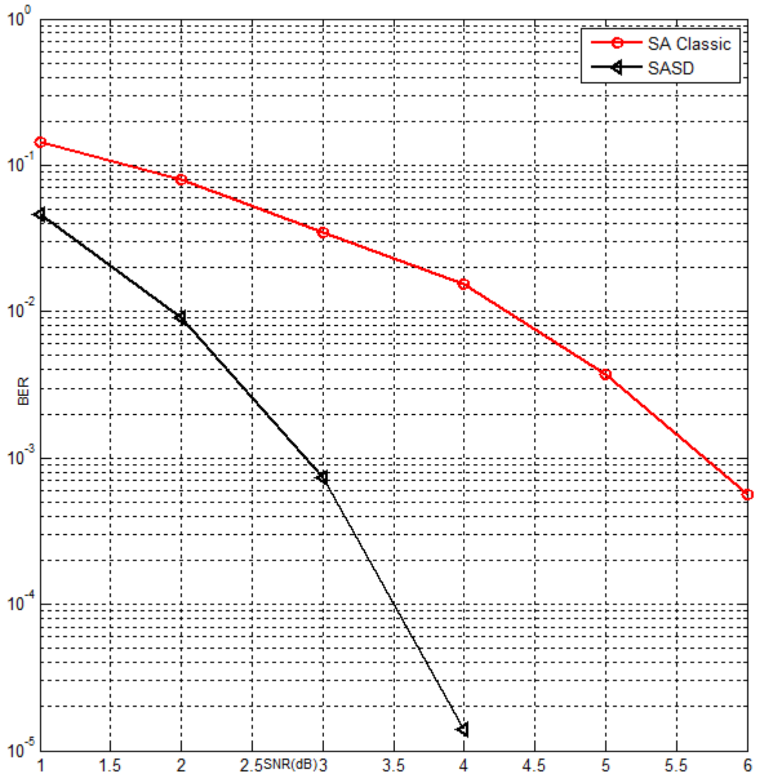

Fig. 8: Code RS(60, 28, 9)

The Fig. 6 and 7 compares the performances of SASD and classical SA for 2 BCH codes. We notice the superiority of our algorithm over the classical algorithm. In fact at 10-4 we have a gain of 2 dB using SASD for BCH (63,45,7) and 1dB for BCH(31,21,5).

The Fig. 8 presents the performances of SASD and classical SA for RS(60,28,9) code. According to this figure, we remark that our algorithm outperforms the classical decoder by 3 dB at 10-3.

In the Fig. 9, our decoder presents a gain of about 2 dB at 10-3 compared to classical SA.

For the LDPC (60,30), the Fig. 10, shows that our algorithm over performs classical SA. According to this figure, we remark that we have about 2.5 dB gain at 10-3.

Comparison of the Proposed Algorithm Versus other Decoders

In this subsection, we compare our SASD decoder with the some known and efficient decoders: Chase-2, OSD-1(Fossorier and Lin, 1995), Maini, SDGA, SIHO, (Chana *et al*., 2011), DDGA, AutDAG, cGA-HP, cGA-M, CGAD and GAMD. For this purpose we run our simulations on several codes as a base of comparison, these simulations where made with default parameters outlined in Table 2. The performances are given in terms of BER (bit error rate) as a function of SNR (Signal to Noise Ratio Eb/N0).

The Fig. 11, shows that our algorithm performs better than AutDAG, SDGA, cGA-HSP, cGA-M and slightly the most efficient decoder DDGA for medium and low noise level.

In Fig. 12, the SASD has better performance than Chase-2, OSD-1, SDGA and SIHO for medium and low noise level. The Maini decoder is more efficient than our decoder for high noise level, however starting from 3.5dB SASD has a little edge over Maini.

The performance of SASD, Chase-2, CGAD, Maini and DDGA algorithms, for the code BCH (31,21,5) code, is shown in Fig. 13.

We notice that our algorithm has almost same performance as Maini and DDGA.

The Fig. 14 compares the performance of our decoder with other decoders for BCH(63,51,5) code. From this figure, we see that our algorithm is the most efficient. Besides at 10-5 it has a gain of 1dB over Chase-2.

Simulations of the non-binary RS(15,7,9), in Fig. 15, reveal the decoding power of SASD over the most efficient decoder DDGA, in fact at 10-5 we have a gain of 1dB, in other hand at 10-3 we gain about 2.5dB over SDGA and CGA-M. In Fig. 15, our algorithm is definitely more efficient than Chase-2, CGAD and cGA-HSP. In fact at 10-3 our decoder has a gain of 3dB over Chase-2 and at 5. 10-3 using SAD we gain about 2dB compared to cGA-HSP.

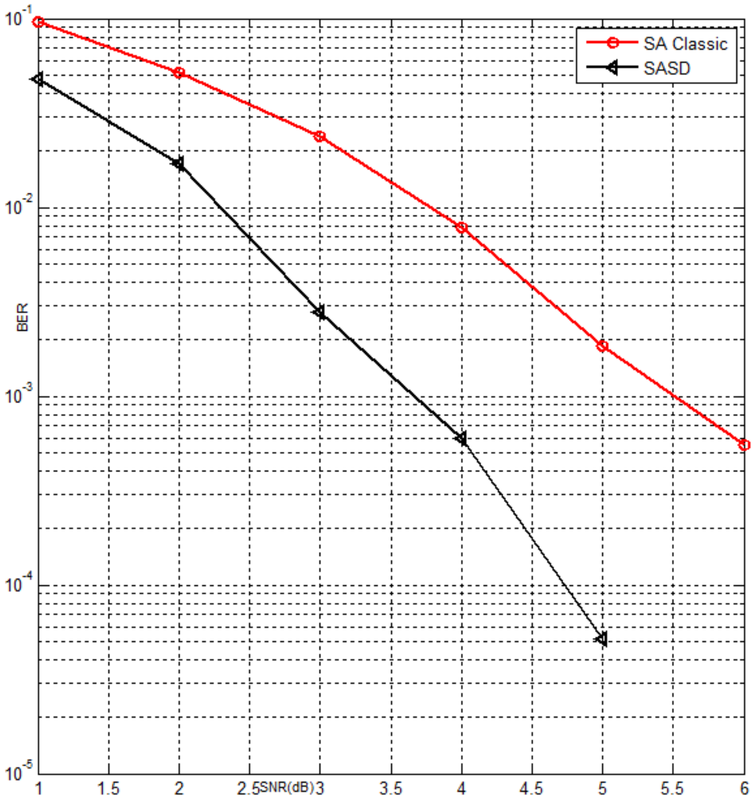

Fig. 9: Code RM(32,16,8)

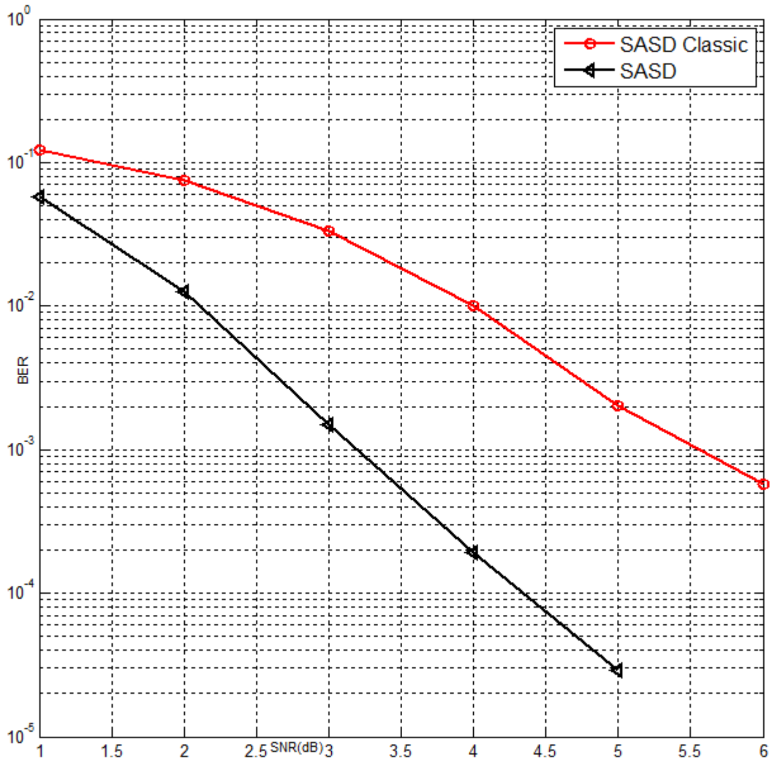

Fig. 10: Code LDPC(60, 30)

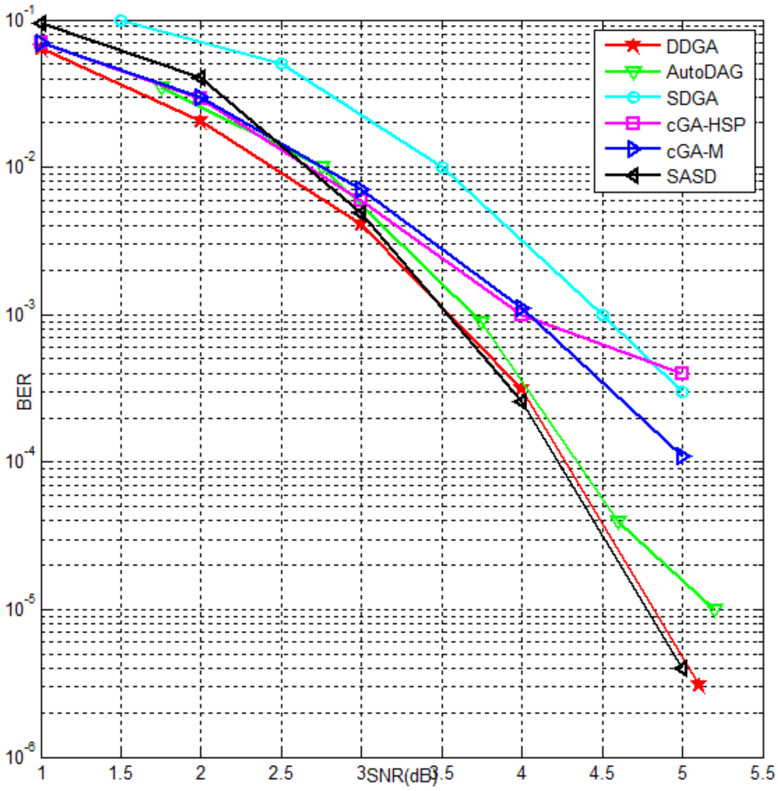

Fig. 11: Code BCH (63, 45, 7)

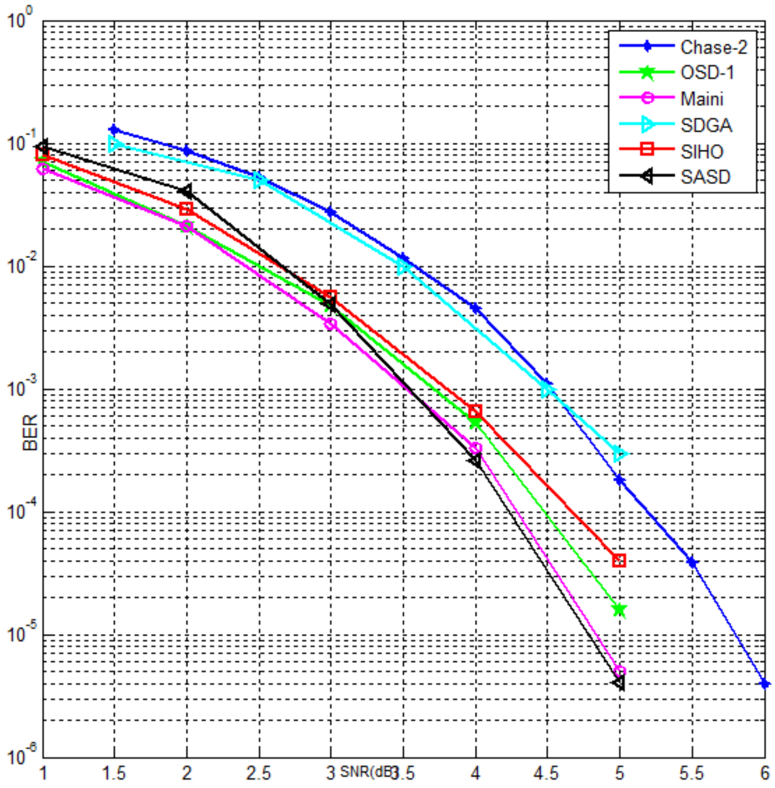

Fig. 12: Code BCH(63,45)

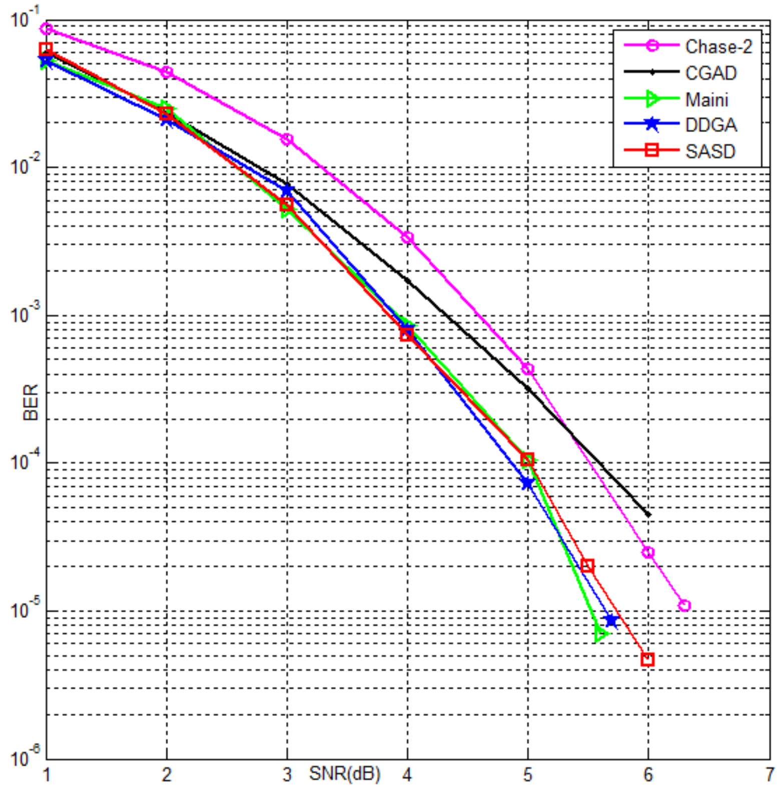

Fig. 13: Code BCH(31,21,5)

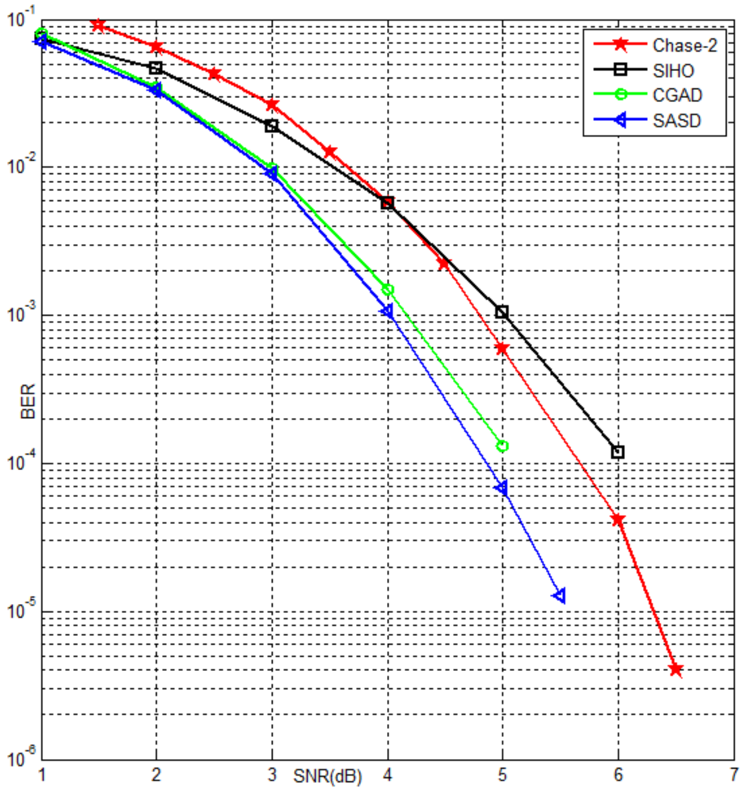

Fig. 14: Code BCH(63,51,5)

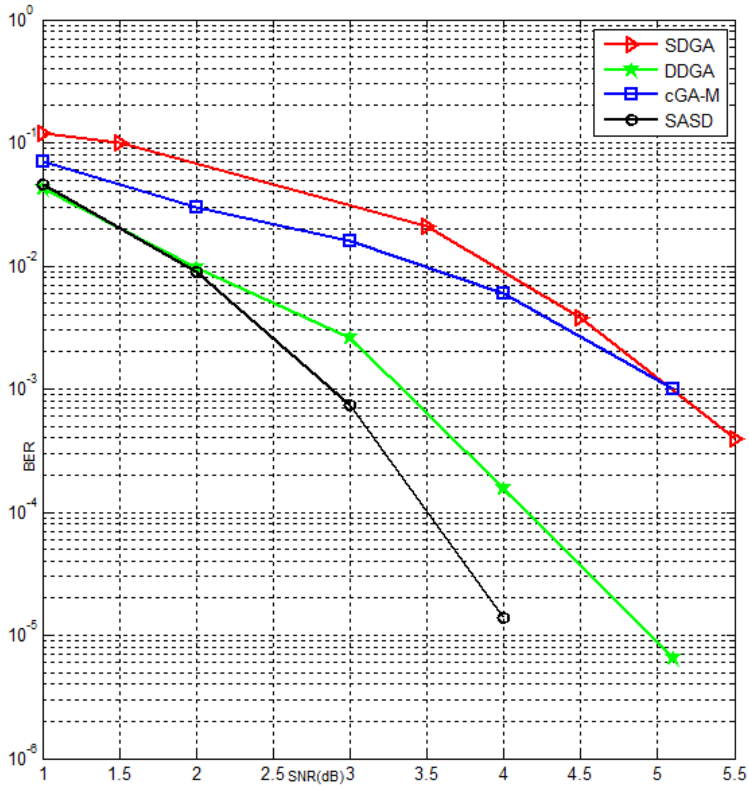

Fig. 15: Code RS(15,7,9)

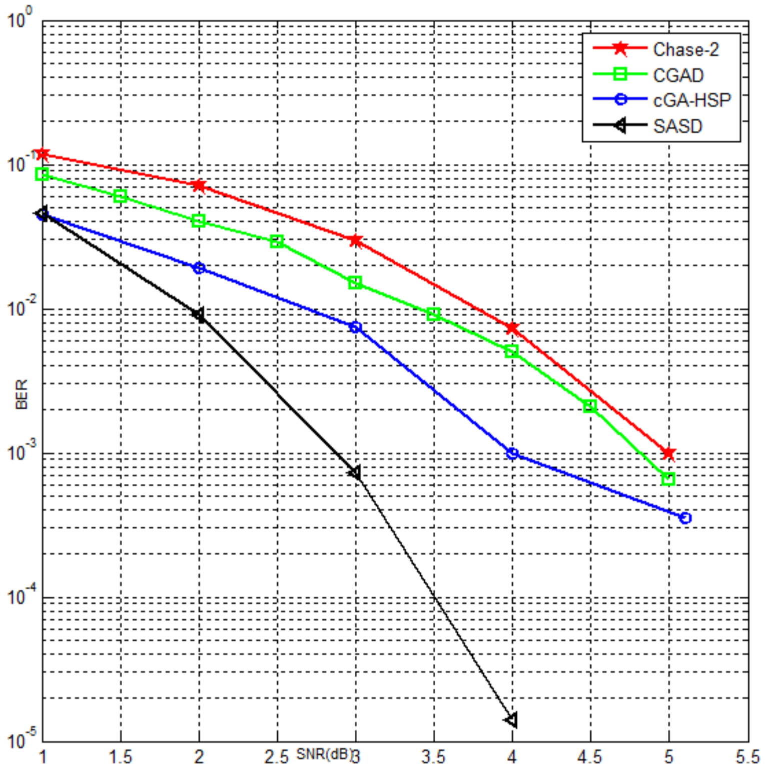

Fig. 16: Code RS(15,7,9)

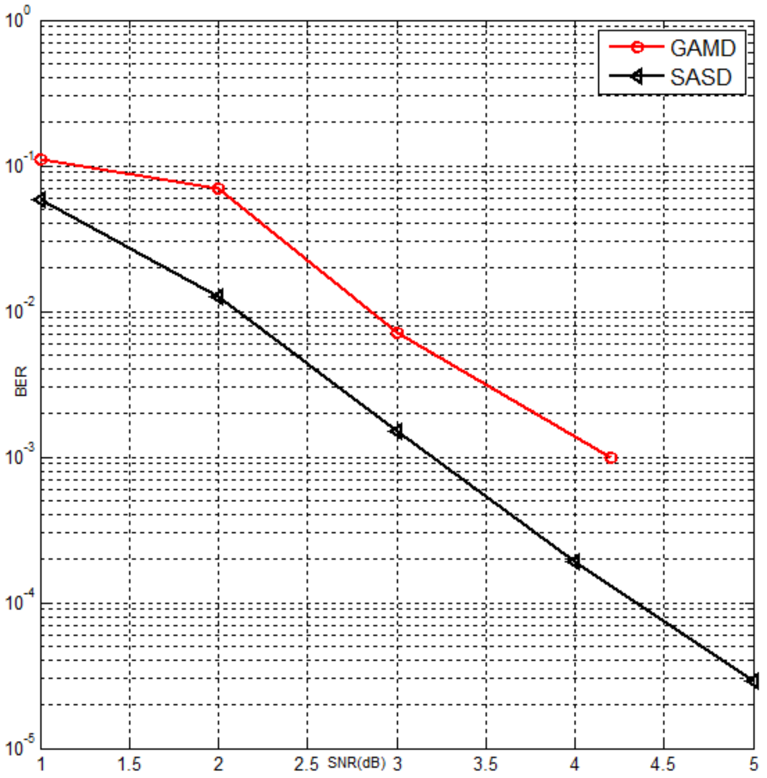

Fig. 17: Code LDPC(60,30)

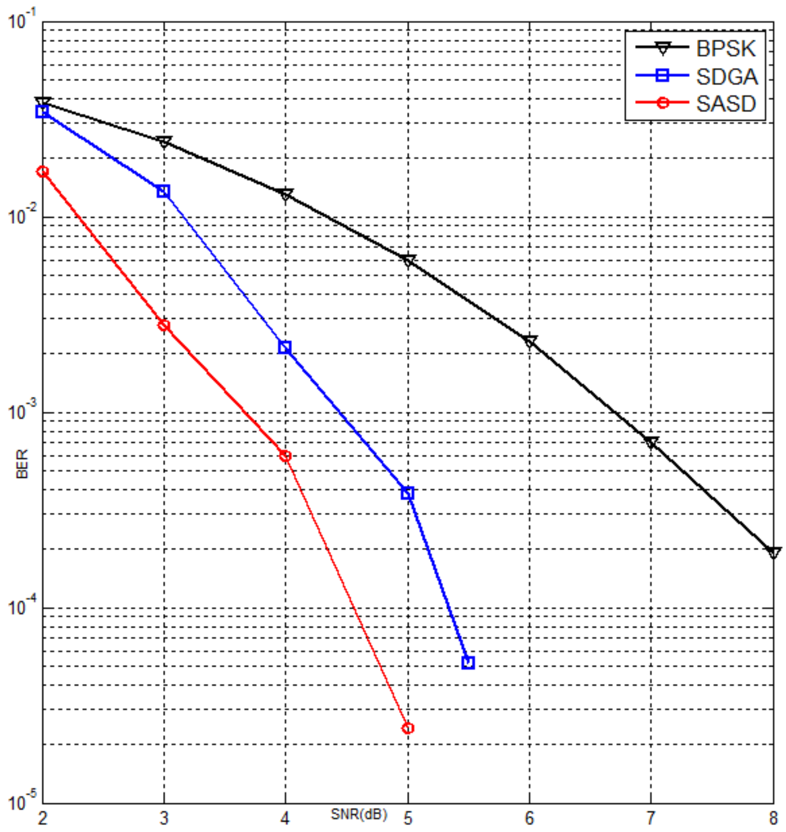

Fig. 18: Code RM(32,16)

The above Fig. 17, shows that for the above LDPC, SASD is applicable for non-cyclic codes and is indeed better than GAMD, actually we can gain about 1 dB by using SASD.

The Fig. 18 compares the performances of SASD, SDGA and the classical BPSK decoding for the non-cyclic RM(32,16) code. We notice the superiority of SASD over the other algorithms. Using our algorithm at 10-4, we have a gain of 0.5 dB over SDGA and about 3.5 dB over BPSK decoding.

Influence of Error Correcting Capability

In order to enhance our algorithm we used the code error correcting capability t as a stop criterion, we will compare in this section the influence of the parameter t, below the algorithm without involving this enhancement:

Parameter Initialization
Ni: number of iterations
Ts: starting *temperature*
Tf: final temperature
a: cooling ratio
S_0: start solution

```
T←Ts
S←S0
While (T >Tf)
{
While (iteration <Ni)
   {
   s_n←pick_neighbor();
      Evaluate dE = E(s_n) – E(s);
   IfdE≤ 0 then s←s_n;
   else if random(0,1) ≤ Exp(-dE/T) then s←s_n;
   end if
   iteration←iteration+1
   }
T←a*T;
}
```

To measure the gain we get from introducing the parameter t we made simulations of the number of explored codewords for both algorithm (with t and without t) in different SNR values, the Fig. 19 summarizes the results.

We can notice that for small and medium noise level, the search space is greatly reduced for example for SNR = 4, we reduce our space by 88%.

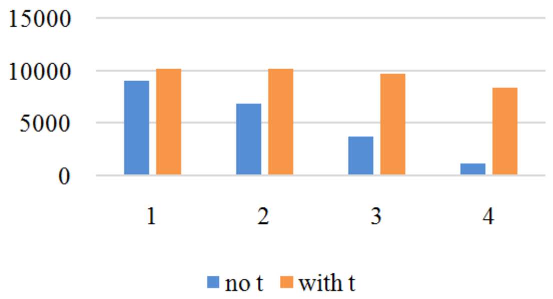

Fig. 19: Codewords for both algorithms (with t and without t)

Table 3: Comparison of algorithm complexity with the concurrent decoders

Algorithm	Complexity
Chase-2	$O(2^t n^2 \log n)$
OSD-1	$O(n^2)$
DDGA	$O(N_i N_g[k(n\text{-}k) + \log(N_i)])$
AutDAG	$O(N_i N_g kn)$
SDGA	$O(2^t(N_i N_g[kn^2 + kn + \log(N_i)]))$
CGAD	$O(T_c k(n\text{-}k))$
Chana dec.	$O(2^{p+1}(k \log n[n + \log(n\text{-}k)]))$
Maini	$O(N_i N_g[kn + \log N_i])$
SASD	$O(N_i N_c kn)$

Complexity Analysis

The Table 3 recapitulates our algorithm complexity with the concurrent decoders.

Let a linear code $C(n,k)$ and t be the error correction capability of the code C, N_i be the population size which is the total number of individuals in the population, N_g be the number of generations and let LNg be the number of generations of local search for Genetic Algorithm, while N_i represents the number of iterations for SASD, N_c is a constant which represents the number of iterations needed for SASD to reach final temperature T_f starting from T_s. T_c a parameter used in CGAD called the average number of generations. The parameter p in Chana decoder is the number of test sequences.

The Chase-2 and SDGA algorithms increase exponentially with t and OSD increases with p, hence for code with large error capability they have the worst complexity. However in terms of code sizes, SASD, DDGA, CGAD, Maini and AutDAG algorithms, the complexity is linear either in k or n. OSD-1, SDGA, Chana and Chase-2 have poor performance in terms of n.

Conclusion

In this study, we have proposed a novel Simulated Annealing based algorithm for soft decision decoding, as far as we know our decoder is the first soft algorithm based on SA. Then we focus on parameter tuning throw several experiments and later show the superiority of our proposed algorithm over the classical SA and over the most famous and up to date decoders, in fact it beats the DDGA by 2dB at 10-5. Furthermore we investigate the enhancement introduced by the error correcting capability parameter in our decoder and we show that we can reduce the search space by 88% for high noise level. Besides, the main advantage of the SASD is that it take advantage of the information reliability of the received word to start search and to generate neighbor solutions, in other side our decoder can be used for non-cyclic and non-binary codes without any complex change. We also compare the algorithmic complexity of our decoder with its concurrent.

The obtained results let us hope to implement this decoder for dual code and try to find other neighbor's search strategies to enhance performance.

Author's Contributions

Lahcen Niharmine, Hicham Bouzkraoui and Ahmed Azouaoui: Contribute on the implementation of the decoder SASD.

Youssef Hadi: Review and correct the article research.

Ethics

I testify that my research paper submitted to the Journal of Computer Science, title: Simulated Annealing Decoder for Linear Block Codes not been published in whole or in part elsewhere. This research project was conducted with full compliance of research ethics norms of Ibn Tofail University, Kenitra, Morocco.

References

Aarts, E. and J. Korst, 1989. Simulated Annealing and Boltzmann Machines. 1st Edn., J. Wiley and Sons, ISBN-10: 0471921467, pp: 272.

Aylaj, B. and M. Belkasmi, 2014. New simulated annealing algorithm for computing the minimum distance of linear block codes. Adv. Comput. Res., 6: 153-158.

Aylaj, B. and M. Belkasmi, 2015. Simulated annealing decoding of linear block codes. Proceedings of the Mediterranean Conference on Information and Communication Technologies, (ICT' 15), Springer, Cham, pp: 175-183. DOI: 10.1007/978-3-319-30301-7_19

Azouaoui, A. and M. Belkasmi, 2012. A new genetic decoding of Linear Block Codes. Proceedings of the International Conference on Multimedia Computing and Systems, May 10-12, IEEE Xplore Press, Tangier, Morocco, pp: 1176-1182. DOI: 10.1109/ICMCS.2012.6320254

Azouaoui, A., A. Berkani and M. Belkasmi, 2012a. An efficient soft decoder of block codes based on compact genetic algorithm. Int. J. Comput. Sci., 9: 431-438.

Azouaoui, A., M. Belkasmi, A. Farchane, 2012b. Efficient dual domain decoding of linear block codes using genetic algorithms. J. Electrical Comput. Eng., 2012: 503834-503834. DOI: 10.1155/2012/503834

Azouaoui, A., M. Belkasmi and A. Farchane, 2012c. Efficient information set decoding based on genetic algorithms. Int. J. Commun. Netw. Syst. Sci., 5: 423-429. DOI: 10.4236/ijcns.2012.57052

Ben-Ameur, W., 2004. Computing the initial temperature of simulated annealing. Comput. Optimiz. Applic., 29: 369-385. DOI: 10.1023/B:COAP.0000044187.23143.bd

Berkani, A., A. Azouaoui, M. Belkasmi and B. Aylaj, 2017. Improved decoding of linear block codes using compact genetic algorithms with larger tournament size. Int. J. Comput. Sci., 14: 15-24. DOI: 10.20943/01201701.1524

Berlekamp, E.R., R.J. McEliece and H.C.A.V. Van, 1978. On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory, 24: 384-386. DOI: 10.1109/TIT.1978.1055873

Cardoso, F.A. and D.S. Arantes, 1999. Genetic decoding of linear block codes. Proceedings of International Conference on Telecommunications Congress on Evolutionary Computation, Jul. 6-9, IEEE Xplore Press, Washington, DC, USA, pp: 2302-2309. DOI: 10.1109/CEC.1999.785561

Chana, I., H. Allouch and M. Belkasmi, 2011. An efficient new soft-decision decoding algorithm for binary cyclic codes. Proceedings of the International Conference on Multimedia Computing and Systems, Apr. 7-9, IEEE Xplore Press, Ouarzazate, Morocco, pp: 823-828. DOI: 10.1109/ICMCS.2011.5945580

Chase, D., 1972. A class of algorithms for decoding block codes with channel measurement information. IEEE Trans. Inform. Theory, 18: 170-182. DOI: 10.1109/TIT.1972.1054746

De Bona, A.A. and W.G. Junior, 2015. Applying hybrid simulated annealing algorithm to the information sets search problem. Wseas Trans. Commun., 14: 381-389.

El Gamal, A., A. Hemachandra, I. Shperling and V.K. Wei, 1987. Using simulated annealing to design good codes. IEEE Trans. Inform. Theory, 33: 116-123. DOI: 10.1109/TIT.1987.1057277

Forney, G.D., 1996. Generalized minimum distance decoding. IEEE Trans. Inform. Theory, 12: 125-131. DOI: 10.1109/TIT.1966.1053873

Fossorier, M.P.C. and S. Lin, 1995. Soft-decision decoding of linear block codes based on ordered statistics. IEEE Trans. Inform. Theory, 41: 1379-1396. DOI: 10.1109/18.412683

Hajek, B., 1988. Cooling schedules for optimal annealing. Math. Operat. Res., 13: 311-329. DOI: 10.1287/moor.13.2.311

Hoffman, J.R., M.M Tenório, E.C.G Wille and W. Godoy Jr, 2011. Applying genetic algorithms to the information sets search problem. Proceedings of the IEEE Third Latin-American Conference on Communications, Oct. 24-26, IEEE Xplore Press, Belem do Para, Brazil, pp: 1-5. DOI: 10.1109/LatinCOM.2011.6107391

Kirkpatrick, S., C.D. Gelatt and M.P. Vecchi, 1983. Optimization by Simulated Annealing, Science, 220: 671-680. DOI: 10.1126/science.220.4598.671

Lee, P.J. and E.F. Brickell, 1988. An observation on the security of McEliece's public-key cryptosystem. Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, (ACT' 88), Springer, Davos, Switzerland, pp: 275-280. DOI: 10.1007/3-540-45961-8_25

Leon, J.S., 1988. A probabilistic algorithm for computing minimum weights of large error-correcting codes. IEEE Trans. Inform. Theory, 34: 1354-1359. DOI: 10.1109/18.21270

MacKay, D.J.C., 2003. Information Theory, Inference and Learning Algorithms. 1st Edn., Cambridge University Press, ISBN-10: 0521642981, pp: 640.

Maini, H.S., K.G. Mehrotra, C. Mohan and S. Ranka, 1994. Genetic algorithms for soft decision decoding of linear block codes. J. Evolut. Comput., 2: 145-164. DOI: 10.1162/evco.1994.2.2.145

Muxiang, Z. and M. Fullong, 1994. Simulated annealing approach to the minimum distance of error-correcting codes. Int. J. Electron., 76: 377-384. DOI: 10.1080/00207219408925934

Nouh, S., I. Chana and M. Belkasmi, 2013. Decoding of block codes by using genetic algorithms and permutations set. Int. J. Commun. Netw. Inform. Security, 5: 201-209.

Prange, E., 1962. The use of information sets in decoding cyclic codes. IRE Trans. Inform. Theory, 8: 5-9. DOI: 10.1109/TIT.1962.1057777

Proakis, J.G., 1995. Digital Communications. 3rd Edn., McGraw Hill, New York, ISBN-10: 0071138145, pp: 928.

Scandurra, A.G., A.L. Dai Pra, L. Arnone, L. Passoni and C. Moreira, 2006. A genetic-algorithm based decoder for low density parity check codes. Latin Am. Applied Res., 36: 169-172.

Shakeel, I., 2010. GA-based soft-decision decoding of block codes. Proceedings of the 17th International Conference on Telecommunications, Apr. 4-7, IEEE Xplore Press, Doha, Qatar, pp: 13-17. DOI: 10.1109/ICTEL.2010.5478650

Shannon, C.E., 1948. A mathematical theory of communication. Bell Syst. Tech. J., 27: 379-423. DOI: 10.1002/j.1538-7305.1948.tb01338.x

Stern, J., 1989. A method for finding codewords of small weight. Proceedings of the 3rd International Colloquium on Coding Theory and Applications, Nov. 02-04, Springer, Berlin, pp: 106-113. DOI: 10.1007/BFb0019850