International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 6 (2018) pp. 4094-4101


# Dual Soft Decoding of Linear Block Codes using Ant Colony Optimization

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

[1]*Department of Computer Science, Faculty of Sciences, Chouaib Doukkali University, El Jadida, Morocco.*
[2]*MISC Laboratory, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco.*
[3]*MISC Laboratory, Faculty of Sciences, Ibn Tofail University, Kenitra, Morocco.*

**Abstract**

In this paper we introduce a novel and efficient soft decoding algorithm for linear block codes. It is known that this decoding is a NP-hard problem where exhaustive search methods become almost impossible in large problem space. After the emergence of metaheuristic approaches, several algorithms were proposed based on these technics and show good performance results comparing to computational resources cost. In this context we present a new soft decoding decision algorithm, based on Ant Colony Optimization, this approach has proven its efficiency in optimization problems. In contrast to existing decoders, the proposed Dual Ant Colony Optimization Soft Decoder (DACOSD) operates on the dual code instead of the code itself. Hence we can optimize our decoder for codes with high rates. This algorithm was simulated over an AWGN channel, then we proceed by tuning of our algorithm parameters, after that, we compare the obtained performances with competitor decoding algorithms. In fact, the proposed decoder is on the top of the most recent soft decoders such DDGA, CGAD-M and CGAD-HSP algorithms. Besides, we discussed the complexity of the proposed algorithm compared to the most known decoders.

**Keywords:** Ant Colony Optimization, Soft Decoding, Error Correcting Codes, Linear Codes, Dual Code, BCH, QR, RS

## INTRODUCTION

Coding theory is interested in finding the best way to transmit information accurately from a source to a destination over different kinds of channels. In real world, the channel is generally perturbed by a noise dependent on environment conditions, hence information reliability is lost. Trying to slightly improve communication performance using classical approaches either by extending hardware capacity or using better material quality, is highly expensive. However, contrary to what was generally believed, Shannon proved in 1948, that there exist a coding scheme which permits transmission that nears the channel capacity with arbitrarily small errors [1]. Thus, software optimization can reduce such cost, actually several algorithms were proposed to enhance communication reliability, with remarkable performance in order to achieve Shannon limit.

To simplify the study, scientists use the model in Fig.1, which depicts different transmission chain components. In fact spotlight was put on decoder part where significant efforts were made in order to find coding/decoding models that approach channel capacity, especially using AWGN channel model. Decoding schemes are classified into two classes, hard and soft decision algorithms. The first approach makes hard decision on the received digital message, thus works only on binary vectors. While Soft-decoding algorithms use digital signal information received directly from demodulation and use real numbers associated with each codeword symbol. Therefore this decoding is capable of correcting more errors.

Soft-decoding is classified as a NP-hard problem [2], where classical algorithms search would not be effective and almost impossible. To tackle this problem, several works were proposed; G.Forney designed a soft decoding algorithm, called “Generalized Minimum Distance Decoding” GMD [3]. OSD Algorithm was also used to implement a soft decision decoder [4]. Nowadays, Artificial Intelligence (AI) and metaheuristic methods were introduced in this field, after their success on optimization problems, several researches were published of decoders based on neural networks and Genetic Algorithms (GA). In fact Maini et al. was the first to introduce a GA decoder [5], and then several GA-Based Decision Soft decoders came after [6-10]. In other side, other algorithms were proposed using dual code [11-13], then recently also Maini was extended to DDGA decoder[14] and two dual soft decoders based on compact GA were proposed in 2017 [15]. Using dual code for high rate codes, seems to reduce considerably the execution time, thus, taking advantage from this simple idea we propose in this paper, a novel soft decoder based on Ant Colony Optimization (DACOSD), as far as we know, this approach has not been previously used to decode linear codes.The remainder of this article will be organized as follow: in section2, we will describe the Ant Colony Optimization method, then in section3 we will explain our DACOSD algorithm. In the section 4 we will discuss the parameter tuning of our decoder then use some famous codes samples base to compare with competitor decoders. Finally we finish our paper by summary of this work and the future trends.

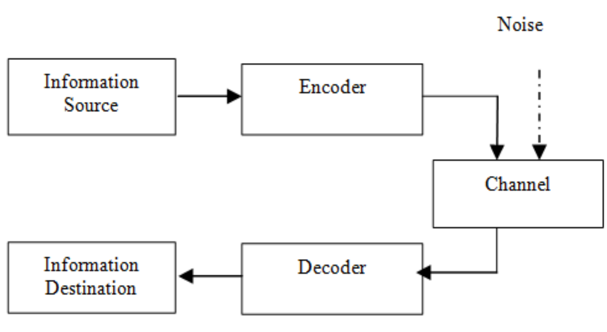


**Figure 1.** Communication system model

## ANT COLONYOPTIMIZATION (ACO)

Ant Colony Optimization algorithms were inspired from ant's behavior when searching food near the nest, they are considered among metaheuristic methods. Dorigo et al. were the first to propose an algorithm using this technic to find the shortest path within a given graph [16]. In fact, biologists have discovered that ants use a chemical substance called pheromone, these ants drop the pheromone on their way, when arriving to some junction, the ants make a probabilistic choice based on pheromone quantity on each possible path, therefore after several walks, the concentration of pheromone grows on some optimal path. However this mechanism alone can glorify sub optimal or blocked ways, that's why the evaporation of the pheromone trails, takes place by weakening the "nasty paths".

The first ACO algorithm called Ant System (AS) was designed to solve the famous Travelling Salesman Problem (TSP) [16]. The problem is to find the shortest path visiting n cities, each city must be visited only once.This mechanism is modelled by a graph G where the cities are the vertices and the edges are formed by paths between cities.In the AS algorithm there is a set of iterations $y$ $(1 \le y \le y_{max})$, every ant $x$ $(1 \le x \le m)$, scans the graph and build a complete path of n cities, for each ant the relative path between city i and city j depends on several conditions:

1) The set of cities already visited by ant $x$ which is on city i, noted by $\boldsymbol{J_i^x}$
2) $\eta_{ij}$ The visibility of city j to i, it is a measure which let the ants to choose the near city rather than the far ones. Often $\eta_{ij}$ is a decreasing function of the $d_{ij}$ the distance between city i and city j (for example ($\eta_{ij} = \frac{1}{d_{ij}}$).
3) The pheromone quantity dropped on the path between 2 cities i and j at iteration $y$, called intensity and noted $\boldsymbol{\tau_{ij}(y)}$ which define the global attractiveness of parts of the whole path and continually updated by the ants.

Arriving to city i the ant $x$ makes a choice based on the following formula which defines the probability to walk to city j:

$$\boldsymbol{P_{ij}^x(y)} = \begin{cases} \frac{(\tau_{ij}(y))^{\alpha}(\eta_{ij})^{\beta}}{\sum_{l \in J_i^x}(\tau_{il}(y))^{\alpha}(\eta_{il})^{\beta}} \boldsymbol{j \in J_i^x} \\ \boldsymbol{0} \end{cases} \quad (1)$$

α and β are 2 parameters which control the relative role of the intensity $\tau_{ij}(y)$ of pheromone and the visibility $\eta_{ij}$, a tradeoff should be done to play on diversification and intensification of the algorithm behavior

After each iteration $y$ every ant $x$ drops a quantity of pheromone over the walked path, depending on solution quality

$$\Delta\boldsymbol{\tau_{ij}^x(y)} = \begin{cases} \frac{Q}{L^x(y)} \boldsymbol{(i,j) \in T^x(y)} \\ \boldsymbol{0,\ otherwise} \end{cases} \quad \mathbf{(2)}$$

Where the $T^x(y)$ is the path done by ant $x$ on iteration $y$, $L^x(y)$ is its length and Q is a fixed parameter.

The evaporation mechanism comes to complete the algorithm, the pheromone intensity decreases on bad paths based on the following rule:

$$\boldsymbol{\tau_{ij}(y+1) = (1-\rho)\tau_{ij}(y) + \sum_{k=1}^{m} \Delta\tau_{ij}^x(y)} \quad \mathbf{(3)}$$

Where m is the size of ant colony and ρ is the evaporation ratio.

The Algorithm can be briefly summarized in the next pseudo code:

```
For each iteration y=1… ymax
    For each ant x=1… m
        Select randomly a city i0
        For each non visited city i
            Select a city j from Jix using equation (1)
        End For
        Deposit pheromone Δτij^x (y) on the path T^x (y)
        using equation (2)
    End For
    Vaporization of the paths using equation (3)
End For
```

## DACOSD Algorithm

We note **C (n, k, d)** a linear code of length n, dimension k and minimum distance d over the field $\boldsymbol{F_2}$. This code can be described by a k×n matrix **G** called Generator matrix, a message $\boldsymbol{m = \{m_i\}_k^1}$ can be then encoded to a codeword $\boldsymbol{c = \{c_i\}_n^1}$ using the equation:

$$\boldsymbol{c = mG} \quad \mathbf{(4)}$$

We define also a parity check (n-k) ×n matrix noted $\boldsymbol{H}$ which satisfies the following property:

$$\forall\, \boldsymbol{v} \in \boldsymbol{F_2^n}, \boldsymbol{v} \textbf{ is codeword } \boldsymbol{<=> Hv^t = 0} \quad \mathbf{(5)}$$

Suppose we transmit the codeword $\boldsymbol{c = \{c_i\}_n^1}$ using BPSK modulation and let $\boldsymbol{z = \{z_i\}_n^1}$ be the modulated signal over a Gaussian channel with noise $\boldsymbol{n = \{n_i\}_n^1}$, where $\boldsymbol{z = \{z_i\}_n^1}$ and $\boldsymbol{n = \{n_i\}_n^1}$ are independent sequences, $\boldsymbol{n_i \sim N\left(0, \frac{N_0}{2}\right)}$ and **N0** is the noise power spectral density. The received signal $\boldsymbol{r = \{r_i\}_n^1}$ is such that $\boldsymbol{r = z + n}$. We note $\boldsymbol{v = \{v_i\}_n^1}$ the hard decision of $\boldsymbol{r = \{r_i\}_n^1}$, the error syndrome $\boldsymbol{s = \{s_i\}_{n-k}^1}$ could be then expressed as follows:

$$\boldsymbol{s = vH^t} \quad \mathbf{(6)}$$

When the syndrome $\boldsymbol{s}$ is zero, that means, the hard decision is a codeword and there was no error in transmission. But when we have an error in transmission, our decoder tries to find the codeword $\hat{c}$ which maximizes the probability $\boldsymbol{p(c|r)}$ over the code space, known as maximum likelihood decoding (MLD). Using the noise distribution and Bayes theorem, the MLD is equivalent to the following equation:

$$f_{r/z} = \frac{1}{(\pi N_0)^{n/2}} \exp\left(\sum_{i=1}^{n} \frac{-(r_i - z_i)^2}{N_0}\right) \quad (7)$$

From the above equation we can conclude that MLD is equivalent to the minimum euclidean distance, before any hard decision. Hence our fitness function will be:

$$f(c) = \sum_{i=1}^{n} (r_i - z_i)^2 \text{ , where } c = \{c_i\}_n^1$$
$$\text{and } z_i = (-1)^{c_i} \quad (8)$$

Rather than working on code space, we will try to find the error $e$ which was added to our signal, then we can deduct the sent, using the following:

$$c = v + e \quad (9)$$

We will try to find how to construct error **e** by heuristic method using dual property, let's suppose that the parity matrix has the form $H = [AI_{n-k}]$. In this case we have:

$$(v + e)H^t = 0 \iff eH^t = s \quad (10)$$

We define reliable information set as the set of most k reliable positions within the received signal $r = \{r_i\}_n^1$ . Using the reliability information set, error could be then written $e = (e_I, e_J)$ where $I$ is the reliable information set and $J$= {1... n}\ $I$, thus equation (10) becomes:

$$(e_I, e_J)\begin{pmatrix} A^t \\ I_{n-k} \end{pmatrix} = s \iff e_I = e_J A^t + s \quad (11)$$

The above equation allow us to reduce complexity and search space to $F_2^{n-k}$ to generate the error part $e_J$ then using (11) generate $e_I$ and then $c = v + (e_I, e_J)$

Based on the previous analysis, the steps of our algorithm are as follows:

1. Make a hard decision $v = \{v_i\}_n^1$ of the received signal $r = \{r_i\}_n^1$:
$$v_i = \begin{cases} 1, & r_i < 0 \\ 0, & r_i \geq 0 \end{cases}$$
2. Calculate the syndrome $s = vH^t$, if $s = 0$ , then output $v$ and exit, otherwise continue
3. Sort the sequences $r = \{r_i\}_n^1$ in decreasing order based on reliability ($|r_i| > |r_{i+1}|$) to obtain new sequences $r' = \{r'_i\}_n^1$, let's note π, the permutation $r' = \pi^{-1}(r)$, we apply π to $H$ to obtain $H'$ and $z' = \pi^{-1}(z)$. The most k reliable $\{r'_i\}_k^1$ will allow our algorithm to search in the set with high probability of errors.
4. Apply Gaussian elimination to $H'$ to obtain a systematic matrix.
5. We apply the ACO algorithm to obtain the most probable errors $e_J = \{e_i\}_n^{k+1}$ occurred in positions {k+1…n}. This algorithm is detailed afterwards.
6. We form the complete error $e = (e_I, e_J)$ using equation (11)
7. The code $c' = v + (e_I, e_J)$ is related to the H' matrix, thus our estimated transmitted codeword is then

$$\hat{c} = \pi^{-1}(c') \quad (12)$$

The ACO algorithm is modeled by the following graph (in case of n=7 and k=4):

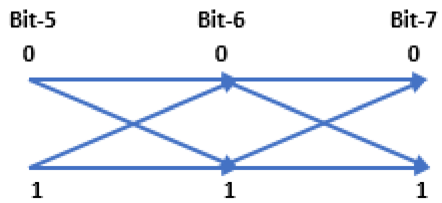


**Figure 1.** Graph of linear code (n=7,k=4)

The search space will be the field $F_2^{n-k}$ represented by the above graph in case of n=7 and k=4.

For ant x at iteration y the probability of setting the bit-i to 1 is defined as follows:

$$P_{i1}^x(y) = \frac{(\tau_{i1}(y))^{\alpha} (\eta_{i1})^{\beta}}{(\tau_{i1}(y))^{\alpha} (\eta_{i1})^{\beta} + (\tau_{i0}(y))^{\alpha} (\eta_{i0})^{\beta}} \quad (13)$$

The parameters α, β, and $\eta_{ij}$ are ACO parameters adapted based on simulation results. At the end of ant x walk all bits $e_i$ are set and the path is the constructed error $e_J = \{e_i\}_n^{k+1}$ , after that the solution fitness is measured by euclidean distance of $z' = \{z'_i\}_n^1$ from $r' = \{r'_i\}_n^1$, thus we define the pheromone to be dropped by ant x at iteration y as follow:

$$\Delta\tau_{ij}^x(y) = \frac{Q}{\sum_{i=1}^{n} (r'_i - z'_i)^2} \quad (14)$$

Where the bit-i ( $k + 1 \leq i \leq n$) is set to $j \in \{0,1\}$ by the ant $x$, and Q is a parameter defined by parameter tuning results. The algorithm could be then summarized as below:

```
Set ACO parameters α, β,ρ, m, Q, y_max
Initialize pheromone intensity τ_ij(0)and visibility η_ij

For each iteration y=1… y_max
    For each ant x=1… m
        Set randomly the bit-(k+1) to "0" or "1"
        For each bit-i i=k+2... n
            Set the bit-i using equation (13)
        End For
        Form the error codeword e using equation (11)
        Compute d_euc = Σ_{i=1}^{n} (r'_i − z'_i)^2
        Update the most probable error e based on best d_euc
        Deposit pheromone Δτ_ij^x (y) on the path T^x (y)
        using equation (14)
    End For
    Vaporization of the paths using equation (3)
End For
```

## RESULTS AND DISCUSSION

### 1. ACO PARAMETERS SETTINGS

We run several simulations of performance (BER) expressed as function of SNR (Signal to Noise Ratio), with different ACO parameters values in order to find the most suitable set to our decoder. For every parameter tuning we fix the other parameters. The remaining parameters were set by default according to the below table:

**Table I:** 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 |

Figure 3, shows, that the best performance is achieved when α=0.1. We can suppose also that the best range for α parameter is [0-1]. Intuitively large values of α tends to amplify the initial paths choosed by ants.

From figure 4, β=2.5, is almost the best choice, we can also use the range [1-3] as the best set of β values, which is confirmed by the recommendation [17].

The performance is quite similar as seen in figure 5, we find $\rho$=0.5 is the most suitable value, which is recommended [17].

From the graph in figure 6, there is no visible best value, by taking the average over all SNR values, it shows that m=50 is the best choice, several simulations prove that it is useless to consider m>k, it is recommended to set $m \approx k$ [17].

In figure 7, The simulations shows that the parameter Q has slight impact on performance, taking into account the average over all SNR values, we may set Q=100.

From theoretic view, the ACO algorithm converges when $y_{max}$ is sufficiently large. However, simulations in figure 8 shows that for $y_{max}$> 500 iterations, there is no visible improvement of performance, therefore we could set $y_{max}$ =500.

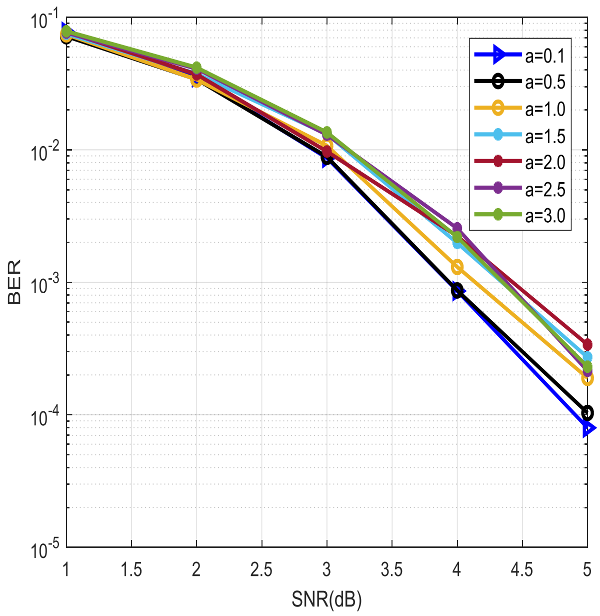


**Figure 3.** Impact of parameter α variation on BER.

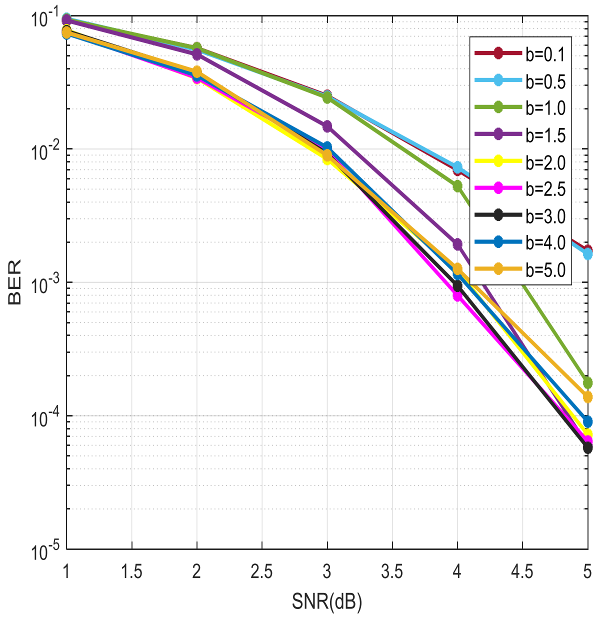


**Figure 4.** Impact of parameter β variation on BER.

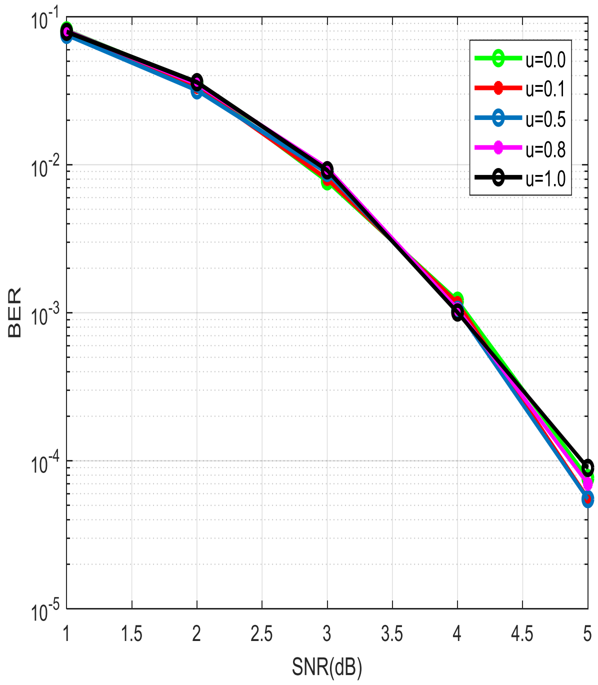


**Figure 5.** Impact of parameter$\rho$ variation on BER.

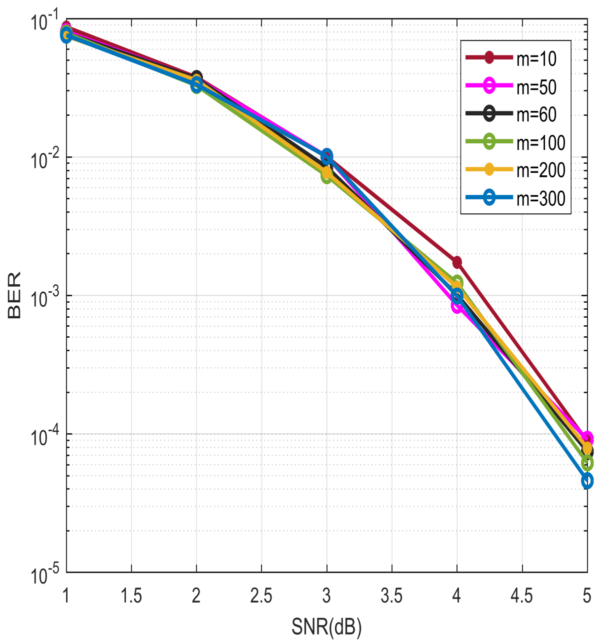


**Figure 6.** Impact of parameter m variation on BER.

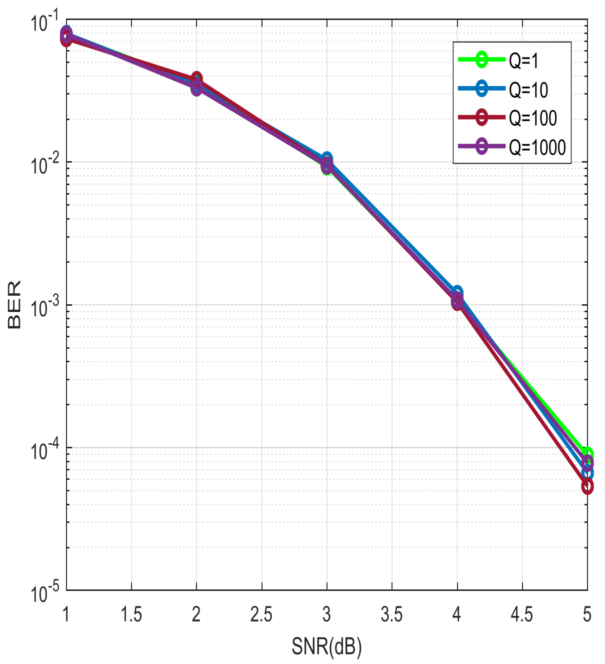


**Figure 7.** Impact of parameter Q variation on BER.

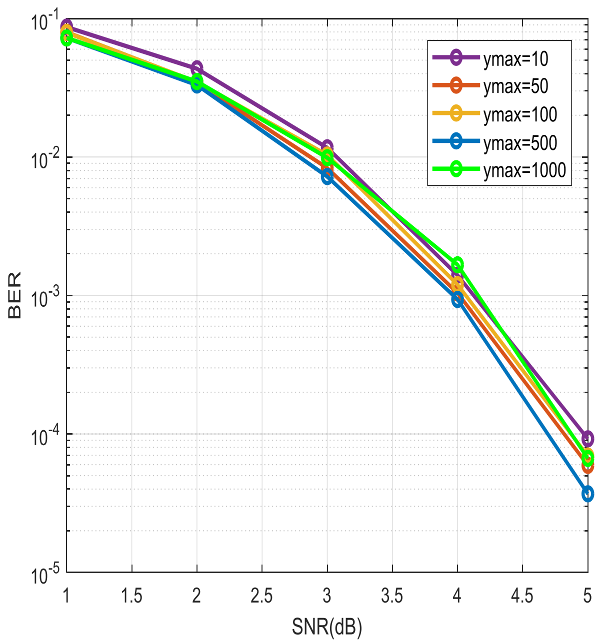


**Figure 8.** Impact of parameter $y_{max}$variation on BER.

## 2. BENCHMARKING

### a) BER Performance

In this subsection, we will show the effectiveness of our dual ACO soft decoder with the some concurrent decoders.

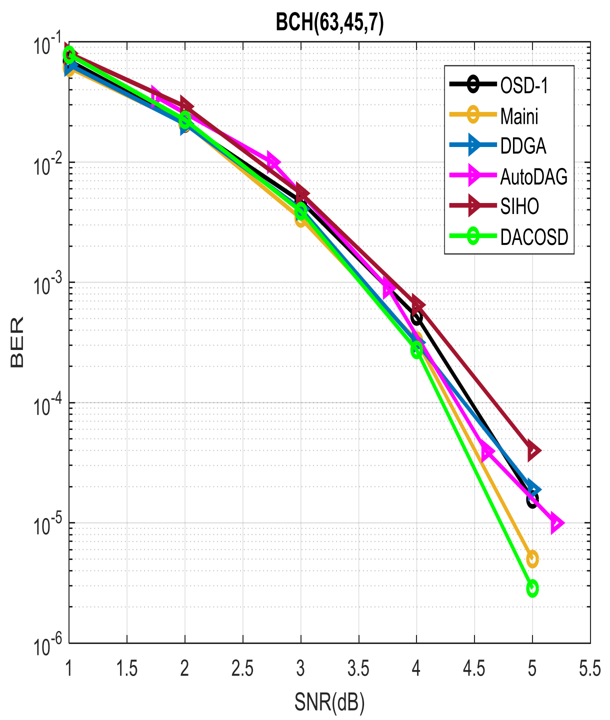


**Figure 9.** Performance of OSD-1,Maini,DDGA, AutoDAG, SIHO and DACOSD decoders on BCH (63, 45, 7)

The figure 9, shows that our algorithm performs better than OSD-1, SIHO [8] and AutDAG [10], but over performs DDGA [14] and Maini [5] only for medium and low noise level

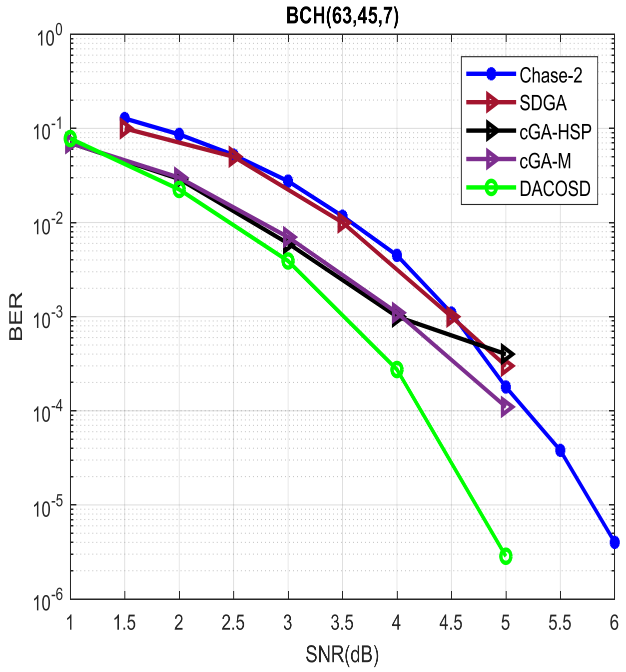


**Figure 10.** Performance of Chase-2, SDGA, cGA-HSP, cGA-M, and DACOSD decoders on BCH (63, 45, 7)

In figure 10, we can see the superiority of the DACOSD over the Chase-2 [18], SDGA [9], cGA-HSP and cGA-M [15]. The gain is 1dB comparing to Chase-2 almost for all BER, at $10^{-4}$ BER the gain is about 0.7dB over the cGA-M.

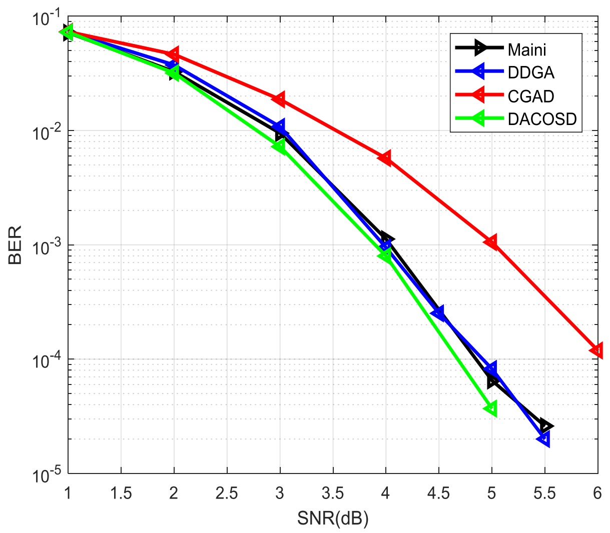


**Figure 11.** Performance of Maini, DDGA, CGAD and DACOSD decoders on BCH (63, 51, 5)

From the figure 11, our decoder performs better than CGAD [11], actually we can have a gain of 1.5 dB at $10^{-4}$, besides, it is slightly better than Maini and DDGA.

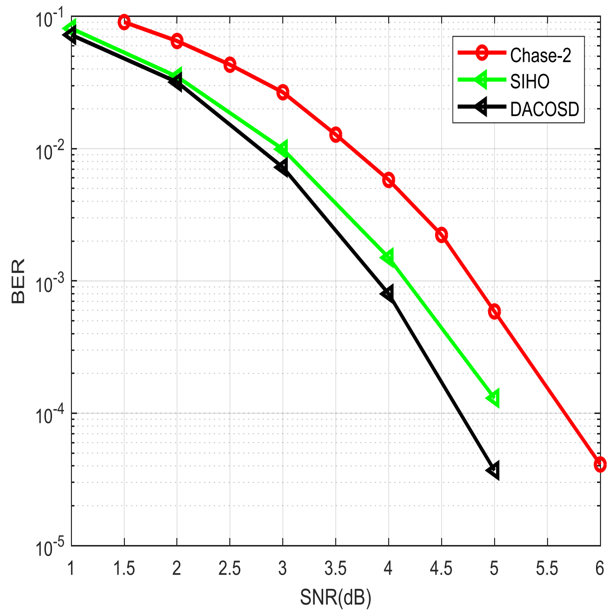


**Figure 12.** Performance of Chase-2, SIHO and DACOSD decoders on BCH (63, 51, 7)

In figure 12, the DACOSD has better performance than Chase-2 and SIHO. In fact at $10^{-5}$ BER we have a gain of 1dB comparing to Chase-2.

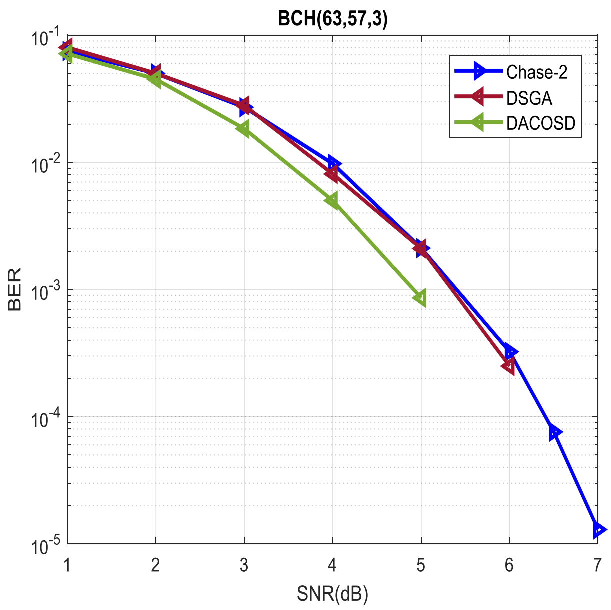


**Figure 13.** Performance of Chase-2, DSGA and DACOSD decoders on BCH (63, 57, 3)

From the figure 13, DACOSD perform better than slightly Chase-2, DSGA [12], the gain is about 0.5dB.

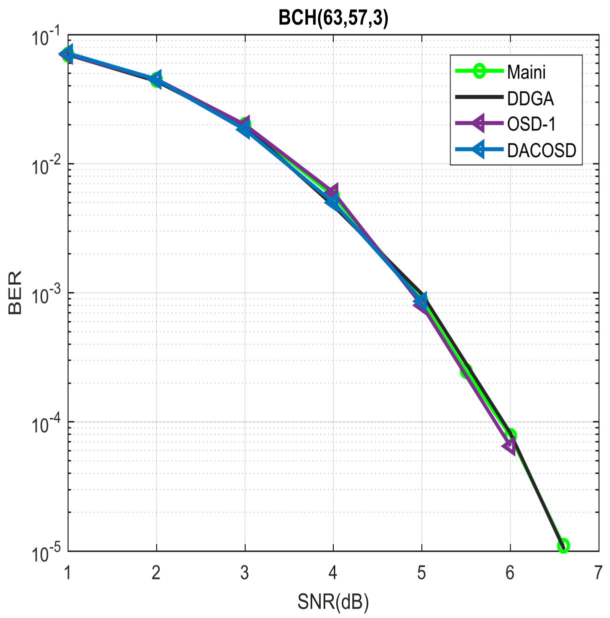


**Figure 14.** Performance of Maini, DDGA, OSD-1, and DACOSD decoders on BCH (63, 57, 3)

The figure 14, show that DACOSD has the same performance as Maini, DDGA and OSD-1.

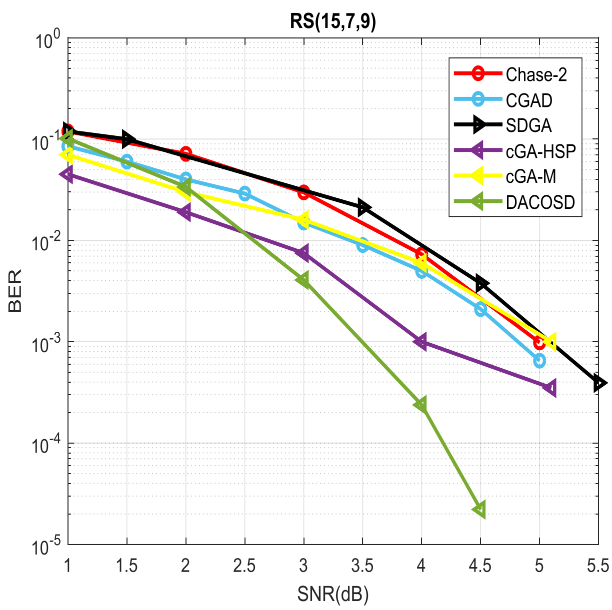


**Figure 15.** Performance of Chase-2, CGAD, SDGA, cGA-HSP, CGA-M and DACOSD decoders on RS (15, 7, 9)

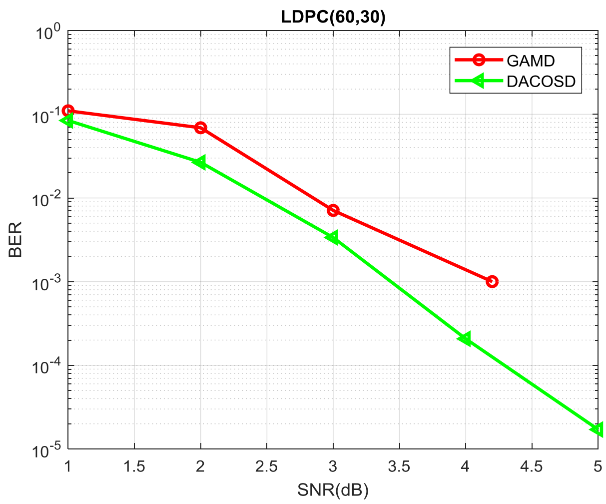


**Figure 16.** Performance of GAMD and DACOSD decoders on LDPC(60, 30)

The above figures 15 and 16, shows that the for RS code, DACOSD is indeed better than Chase-2, CGAD, SDGA and the most up to date decoders cGA-HSP, cGA-M. M. For LDPC code DACOSD is more effective than GAMD [7], we can gain about 0.7 dB at $10^{-3}$BER.

## b) Complexity Analysis

The below table summarize algorithm complexity of differentcompetitor's decoders:

**Table II:** Complexity

| Algorithm | Complexity |
|---|---|
| Chase-2 | $O(2^t n^2 log_2 n)$ |
| Maini | $O(N_i N_g [kn + logN_i])$ |
| DDGA | $O(N_i N_g [k(n-k) + logN_i])$ |
| OSD-1 | $O(n^2)$ |
| SDGA | $O(2^t (N_i N_g [kn^2 + kn + \log(N_i)]))$ |
| CGAD | $O(T_c k(n-k))$ |
| DACOSD | $O(my_{max}(n-k)n)$ |

- ✓ t=[d-1]/2, the error correcting capability
- ✓ Ni, parameter used is Genetic Algorithm, called population size.
- ✓ Ng, parameter used is Genetic Algorithm, called number of generations.
- ✓ Tc, parameter used is CGAD algorithm, called average number of generations.

For codes with high rates($k \sim n$), DDGA, CGAD and DACOSD have linear complexity with n, while Chase-2, Maini, OSD-1 and SDGA have worst complexity, about $O(n^2)$

Using simulations setting, for codes with high rates($\frac{k}{n} \geq 90\%$) the complexity might be upper bounded for DDGA, CGAD and DACOSD as follows

**Table III:** Complexity in (k, n)

| Algorithm | Parameters Setting | Complexity |
|---|---|---|
| DDGA | $N_i = 300, N_g = 100$ | $O(3000k\text{n})$ |
| CGAD | $T_c = 1000$ | $O(100k\text{n})$ |
| DACOSD | $m \approx k, y_{max} = 500$ | $O(50\text{k}n)$ |

From the above table, we can see that for k<200 our algorithm has the lowest complexity comparing to its concurrent decoders for high rate codes.

## CONCLUSION

This paper has made original contribution in the area of soft decoding algorithms by introducing Ant Colony Optimization metaheuristic approach, using dual code instead of the code itself and taking advantage of the reliability information of the received signal, to provide larger coding gain and less algorithmic complexity. The performance of this decoder was investigated with different codes over an AWGN channel, the obtained results confirm that it performs better than its competitors. We focus also on algorithm parameter tuning to enhance the performance. The proposed algorithm can be used for non-cyclic and non-binary codes and on different kind channel models, with low implementation changes.

Although we tested our algorithm on linear block codes, there is no reason why not use it for convolutional codes and combine this approach with different metaheuristic methods to enhance performance.

## REFERENCES

[1] C.E.Shannon, “A Mathematical Theory of Communication”, The Bell System Technical Journal,Vol. 27, pp. 379–423, 623–656, July, October, 1948

[2] E. Berlekamp, "Non-binary BCH decoding" (San Remo, Italy), pp. 459-470, International Symposium on Information Theory, 1967.

[3] G.Forney "Generalized minimum distance decoding", IEEE Transactions on Information Theory (Vol12, Issue: 2, Apr 1966).

[4] M. P. C. Fossorier and S. Lin, “Soft-decision decoding of linear block codes based on ordered statistics,” IEEE Transactions on Information Theory, Vol. 41, no. 5, pp 1379–1396, 1995.

[5] H.S.Maini, K. G. Mehrotra,C. Mohan, S. Ranka, "Genetic Algorithms for Soft Decision Decoding of Linear Block Codes", Journal of Evolutionary Computation, Vol 2, No.2,pp.145-164, Nov.1994.

[6] A. C. M. Cardoso, D. S. Arantes, "Genetic Decoding of Linear Block Codes," Congress on Evolutionary Computation, Washington, DC, USA, 1999.

[7] Adriana G. Scandurra, C. Moreira, "A Genetic-AlgorithmBasedDecoderforLowDensityParityCheckCodes", Latin American Applied Research 2006, Vol 36, No 3, pp 169-172.

[8] I. Chana, H. Allouch and M. Belkasmi, “An efficient new soft-decision decoding algorithm for binary cyclic codes”,IEEE International Conference On Multimedia Computing and Systems (ICMCS’11) proceedings , pp 823-828, Ouarzazate, Morocco, April 2011.

[9] A.Azouaoui, M.Belkasmi, A.Farchane, "Efficient Information Set Decoding Based on Genetic Algorithms", International Journal of Communications, Network and System Sciences, 2012, Vol 5, No 7, pp 423-429.

[10] S.Nouh, I.Chana, M.Belkasmi, "Decoding of Block Codes by using Genetic Algorithms and Permutations Set", International Journal of Communication Networks and Information Security, December 2013, Vol 5, No 3, pp 201-209.

[11] A.Azouaoui, A.Berkani, M.Belkasmi, "An Efficient Soft Decoder of Block Codes Based on Compact Genetic Algorithm", International Journal of Computer Science Issues, September 2012, Vol 9, Issue 5, No 2, pp 431-438.

[12] A.Azouaoui, M.Belkasmi, "A new genetic decoding of Linear Block Codes", in Proceedings of 2012 International Conference on Multimedia Computing and Systems, ICMCS 2012, pp 1176-1182.

[13] I. Shakeel, "GA-based Soft-decision Decoding of Block Codes", IEEE 17th International Conference on Telecommunications, Doha, Qatar, 4-7 April 2010, pp 13-17.

[14] A.Azouaoui, M.Belkasmi, A.Farchane, "Efficient Dual Domain Decoding of Linear Block Codes Using Genetic Algorithms", Journal of Electrical and Computer Engineering, Vol 2012 (2012), Article ID 503834, 12 pages.

[15] A.Berkani, A.Azouaoui, M.Belkasmi, B.Aylaj, “Improved Decoding of linear Block Codes using compact GeneticAlgorithms with larger tournament size”, International Journal of Computer Science Issues, January 2017, Vol 14, Issue 1, pp 15-24

[16] M. Dorigo, L.M. Gambardella, "Ant colonies for the traveling salesman problem", Biosystem 43 (1997) pp 73–81.

[17] M. Dorigo, T.Stutzle, "Ant colony Optimization" (2044) Page 71.

[18] D. Chase, "A Class of Algorithms for Decoding Block Codes with Channel Measurement", 1972.Information, IEEE Trans.Inform. Theory, Vol, 18, pp 170–181, January.