A Genetic Algorithm for Optimizing TCM Encoder

Rekkal Kahina
LTIT Laboratory
Tahri Mohammed University-Bechar
Street independence B.P 417
Bechar, Algeria

Abdesselam Bassou
Department of Electrical Engineering, Faculty of Technology,
Tahri Mohammed University-Bechar
Street independence B.P 417

***Abstract*—This article describes a genetic algorithm for the optimization of the Trellis Coded Modulation (TCM) schemes with a view to achieve a higher performance in the multipath fading channel. The use of genetic algorithms is motivated by the fact that they are capable of performing global searches to retrieve an approximate solution to an optimization problem and if the solution is unknown to provide one within a reasonable time lapse. The TCM schemes are indeed optimized by the Rouane and Costello algorithm but the latter has as major disadvantage high requirements in both computation time and memory storage. This is further exacerbated by an increase in the encoder rate, the number of memory piles and the depth of the trellis. We describe a genetic algorithm which is especially well suited to combinatorial optimization, in particular to the optimization of NP-complete problems for which the computation time grows with the complexity of the problem, in a non-polynomial way. Furthermore this opens up the possibility of using the method for the generation of codes for channel characteristics for which no optimization codes are yet known. Simulation results are presented, that show the evolutionary programming algorithm on several generations of populations which only exhibit a medium probability of exchanging genetic information.**

Keywords*—*Trellis Coded Modulation; free distance; genetic algorithm

I. INTRODUCTION

Multilevel modulation of convolutionally encoded symbols was a technique known before the introduction of TCM. The innovative aspect of TCM is the idea that convolutional encoding and modulation should be treated as a unique operation and not as separate entities [1].

As a result, instead of first demodulating and then decoding the received signal, the demodulation and decoding are combined, by the receiver, in a single process. Consequently, the parameter governing the performance of the transmission system is no longer the free Hamming distance of the convolutional code, but becomes the free Euclidean distance between transmitted signal sequences, over the additive white Gaussian noise channel. Thus the optimization of the TCM design will be based on Euclidean distances rather than on Hamming distances, so that the choice of the code and of the signal constellation will not be performed separately. To this end, much research has been done to optimize TCM encoder schemes by maximizing the free distance as is the case in Rouane and Costello's work [2] which defines an algorithm aiming to maximize the free distance which computes the spectrum of linear, regular and quasi-regular trellis codes.

This optimization has not only affected the design of TCM schemes but has also affected the points of constellations where Matthew C Valenti [3] had described a genetic algorithm for solving the symbol labelling problem and extending the algorithm to optimize their location in the signal space. Yang [4] describes a search method, based on genetic algorithms, to solve the problem of the signal constellations mapping which minimizes the BER of the system. In [5] Confessore describes a genetic algorithm to design satellite constellations for regional coverage. The performance was tested by evaluating optimal satellite configurations both for global coverage and for regional coverage and an extensive series of computational tests was performed to validate the meta-heuristic approach proposed for regional coverage. Tania [6] made use of a genetic algorithm to maximize the percent coverage and minimize the revisit time for a small satellite constellation with limited coverage. Anit Kumar [7] studied the encoding in genetic algorithms which is essentially dependent on the type of problem and examined different coding schemes according to the problems in which they are used.

In fact, genetic algorithms are used by large companies to optimize schedules and design products which can range from large aircrafts to tiny computer chips or to medicines [8]. They thus make use of the power and efficacy of genetic algorithms which are able to find solutions to problems that other optimization methods cannot handle because of a lack of continuity, derivatives, linearity, or other features. Our work aims to optimize the TCM scheme for which we propose a genetic algorithm to design and optimize the placement of branches which are used for the systematic and parity bits and the connectivity of the memories between them. By letting the design evolve over several generations, new encoder formats are found with lower binary error levels than had previously been obtained by applying one of the two modulation techniques QAM or PSK depending on the choice of encoder rate.

The remainder of the paper is organized as follows: Section II provides a model of the system under consideration. Section III provides the simulation results with a comparison of the initial and second population; the best code generated by

the genetic algorithm as compared to the Ungerboeck code in the cases of three, four and five memories. Section IV concludes the paper.

II. System Model

Genetic algorithms are a type of optimization algorithm used to find the optimal solution(s) to a given computational problem that maximizes or minimizes a particular function. In this article, our goal is to maximize the free distance using a genetic algorithm. Before giving the fitness function, we will first provide a definition of free distance.

A. Definition of Free Distance (d_{free})

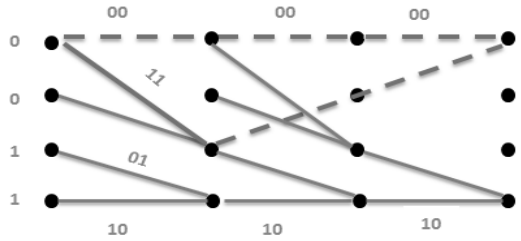


Fig. 1. Two pairs of paths diverging at time n = 0 and reaching the same states at the same time.

Free distance represents the shortest distance between two divergent paths starting from the same point (state) and returning to that point as quickly as possible as shown in Fig.1. This distance represents the distance between two nearest paths without error.

B. Design of the TCM Encoder

The generic TCM encoder, proposed by Ungerboeck [1], is presented in Fig. 2. It allows the generation from m input bits (within the input symbol u_n), an output that contains the m systematic bits and a parity bit produced using a combination of the m input bits and the last state of the encoder memory of size v that yields a coding rate R = $\frac{m}{m+1}$. The design of a TCM encoder consists in the determination of the parameters h_{ij}, i ∈ $\{0,1,..,m\}$ and j ∈ $\{0,1,....v\}$, which can take values from{0,1}. The TCM encoder, presented in (2), imposes the following assumption: $h_{00} = h_{0v} = 1\ and\ h_{i0} = h_{iv} = 0, i \in \{1,..,m\}$.

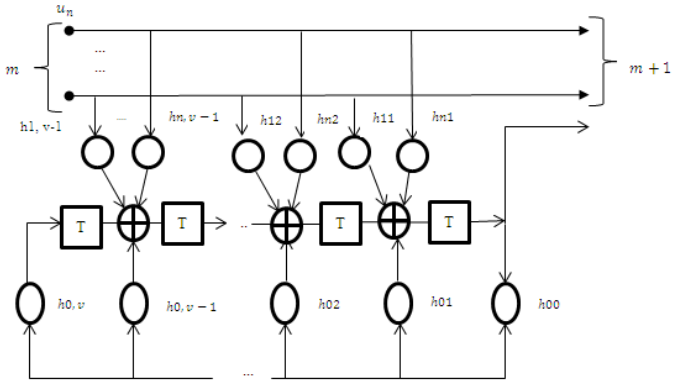


Fig. 2. Generic TCM encoder [1].

The TCM encoder can be represented, using the parameters h_{ij}. by polynomials $H_i(D)$ of the variable D as follows: $H_i(D) = \sum_{j=0}^{v} h_{ij} . D^j$, i ∈ $\{0, ..., m\}$, or by the code generator $(h_0, h_1, ..., h_m)$ written in octal [9].

Let G be a $v \times v$ matrix which describes how the state variables at time n+l are related to the state variables at time n, and let T be a $v \times m$ matrix which describes how the symbols at time n+l depend on the input symbol at time n. Using these relationships, the future state (S_{n+l}) of the encoder in Fig. 2 can be constructed as follows:

$$S_{n+1} = G.S_n + T.u_n \quad (1)$$

Where, u_n is a $m \times 1$ vector describing the current input, and S_n is a $v\ x\ 1$ vector describing the current state.

Equation (1) can be expressed as a polynomial using D-transform as in [10].

$$S(D) = (I.D + G)^{-1}T\ .x(D) \quad (2)$$

- Example on TCM encoder.

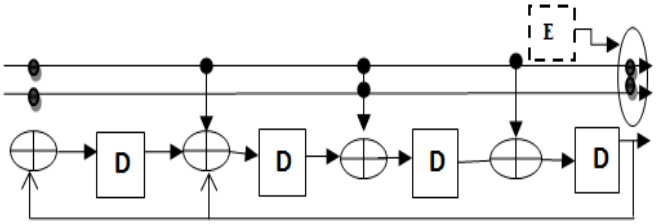


Fig. 3. Example of a 16-state TCM encoder [1].

Consider the 16-state rate 2/3 TCM encoder shown in Fig. 3 where "⊕" represents the operation of XOR and "D" represents the memory or the shift register of the TCM encoder.

The TCM encoder in Fig. 3 has three polynomials expressed as:

$$H_0(D) = D^4 + D + 1$$

$$H_1(D) = D^2$$

$$H_2(D) = D^3 + D^2 + D$$

and, the code generator is ($h_0 = 10011_2 = 23_8$, h1 = $00100_2 = 04_8$, $h_2 = 01110_2 = 16_8$), or in shorthand (23, 04, 16).

The matrix representation of this encoder configuration is given by

$$G = \begin{matrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{matrix} \quad \text{and} \quad T = \begin{matrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{matrix} \quad \text{and E} = 0 \quad 0$$

C. TCM Encoder Representation using a Genetic Algorithm

Fig. 4 represents the TCM encoder with rate 2/3 comprising three branches, two of which represent the systematic bits and

one represents the parity bit. "⊕" represents the operation of XOR and "D" represents the TCM encoder memory.

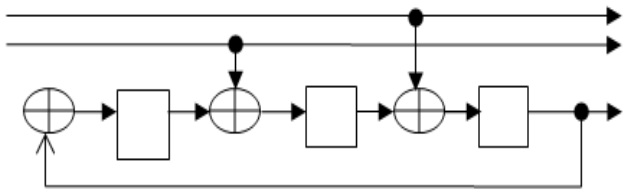


Fig. 4. Example of an 8-state TCM encoder [1].

The TCM encoder in Fig. 4 has three polynomials expressed as: h0 (D) = D^3+1; h1 (D) =D; h2 (D) = D^2, and the generator code is $(h0 = (1001)_2 = (11)_8$, $h1 = (0010)_2 = (02)_8$, $h2 = (0100)_2 = (04)_8$.

The matrix representation of this encoder configuration is as follows:

$$G = \begin{matrix}0&0&1\\1&0&0\\0&1&0\end{matrix}, \quad T = \begin{matrix}0&0\\0&1\\1&0\end{matrix}$$ and the output of the encoder

E = [0 0].

D. Binary Coding of the TCM Chromosome

The term chromosome refers to a numerical value or values that represent a candidate solution to the problem that the genetic algorithm is trying to solve [11]. Each candidate solution is encoded as an array of parameter values, where the performance of a genetic algorithm depends highly on the method used to encode this candidate solution into chromosomes and on what the fitness function is actually measuring [12].

In our work we represent the TCM encoder as an array with binary values which represent the three variables matrix G, T, E Where $G = [001100010]_2 = [98]_{10}$, $T = [001010]_2 = [10]_{10}$, $E = [0\ 0]_2 = 0$.

TABLE I. CHROMOSOME OF THE TCM ENCODER

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	1	0	0	0	1	0	0	0	1	0	1	0	0	0

- The size of the chromosome in Table 1 is equal to 17 bits in the case of three (3) memories, 26 bits in the case of four (4) memories and 37 bits in the case of five (5) memories.
- Some data is handed down by the parents, from one generation to the next. These are the hereditary genes. In our case to keep the same characteristics as for a recursive convolutional encoder the following cell indexes remain unchanged (Table 2) and are inherited by all the new generations of the TCM chromosome.

TABLE II. CELLS INHERITED FROM THE FATHER

3	4	5	7	8
1	1	0	0	1

In this paper, we work on recursive systematic code where matrix G represents the connectivity of the memories between them. In matrix G, value 1 (first row, last column) is set in position G [1, m], which makes the encoder recursive. Matrix G can be represented by 4 possible cases, as follows:

$$G = \begin{matrix}0&0&1\\1&0&0\\0&1&0\end{matrix} \text{ or } \begin{matrix}0&0&1\\1&0&0\\0&1&1\end{matrix} \text{ or } \begin{matrix}0&0&1\\1&0&1\\0&1&0\end{matrix} \text{ or } \begin{matrix}0&0&1\\1&0&1\\0&1&1\end{matrix}$$

In the case of matrix "T", as shown in Fig. 5, there are six (6) possible connexions. The number of possible cases for matrix "T" is equal to $2^{(Ne*m)}$ where "Ne" represents the number of bit per symbol and "m" represents the number of memories.

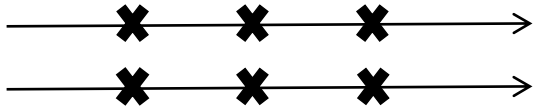


Fig. 5. Branches of systematic bits with six (06) possible connexions.

Finally, vector E represents the output of the TCM encoder, where the output of the encoder takes four candidates as follows: E = [0 0] or [0 1] or [1 0] or [1 1].

E. Reproduction of the TCM Encoder Operations using the Genetic Algorithm

Genetic algorithms can be applied to any process control application for the optimization of different parameters. Genetic algorithms use various operators viz. crossover, mutation for the proper selection of an optimized value. The selection of the appropriate crossover and mutation technique will depend on the encoding method and the problem requirement [13].

- Crossover: Applies to two different individuals where the result is:
  - Chromosome formed from the genes of its two parents.
  - Two children are "produced" for the next generation.
  - The decrement percentage is fixed.

A parent 1:

0	0	1	1	0	0	1	0	0	0	1	0	1	0	0

A parent 2:

0	0	1	1	0	1	1	1	0	0	0	1	0	1	1	0	1

Child 1:

0	0	1	1	0	0	1	1	0	0	0	1	0	1	1	0	0

Child 2:

0	0	1	1	0	1	0	1	0	0	0	1	0	1	0	0	0

- Mutation: Applies to a single individual by modifying one or more genes of the selected parent(s), where the result is:
  - A single new child is produced.
  - A mutation percentage is fixed.

A parent 1:

0	0	1	1	0	0	0	1	0	0	0	1	0	1	0	0	0

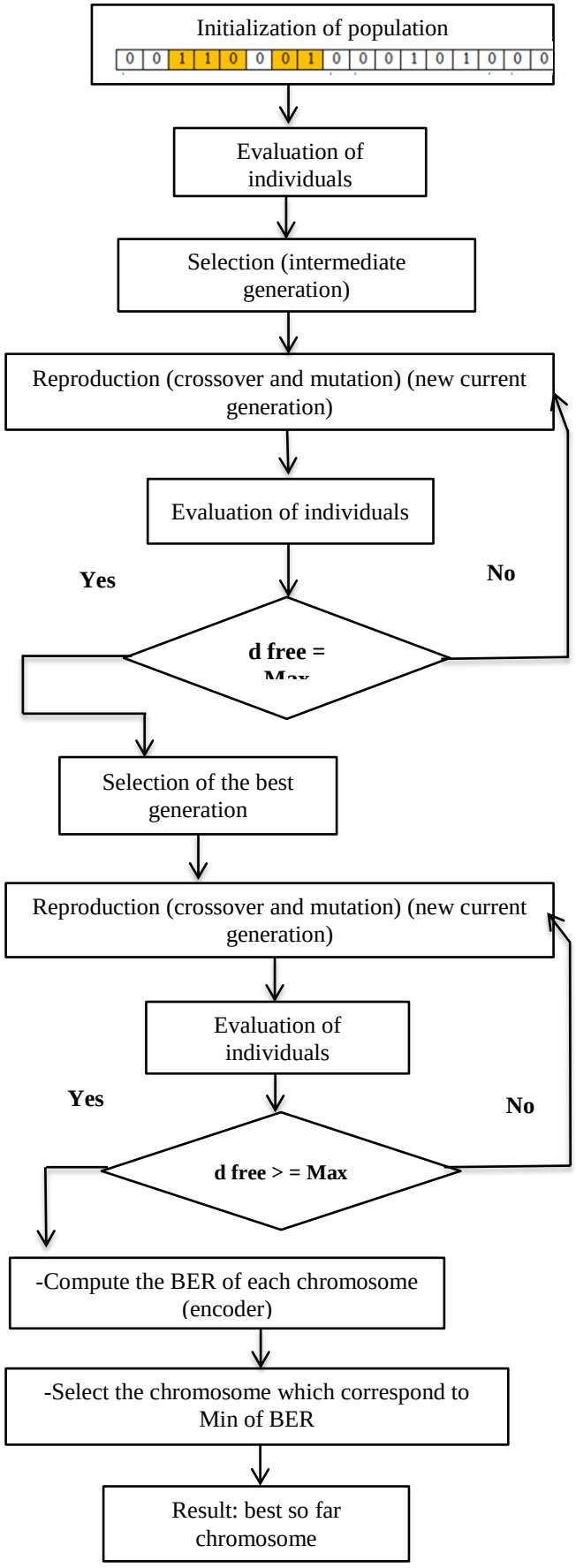
Child 1:

0	0	1	1	0	1	0	1	1	0	1	1	0	1	0	0	0

- Choose the selection method with the following three main objectives:
  - Choice of individuals which apply the reproduction operations for the creation of future generations (creation of a mating-pool).
  - "Selection" of the best individuals.
  - Possibility of exploring the different parts of the research setting.

F. Flowchart of the Genetic Algorithm for the TCM Scheme Optimization

Genetic algorithms begin with a set of solutions represented by chromosomes, called a population. Solutions from one population are taken and used to form a new population, which is motivated by the possibility that the new population will be better than the old one. Furthermore, solutions are selected according to their fitness to form new solutions, i.e. offsprings. The above process is repeated until some given condition is satisfied. The basic genetic algorithm is outlined as below:



III. Simulation Results

In this work, the genetic algorithm used to optimize the TCM encoder, based on maximizing the free distance of quasi-regular trellis code with three variables (G, T, E) in the interval [0, 5] where the fitness function is:

Max [F = MyGa (fitness, G, T, E)]

The algorithm is initialized with a population of 15 chromosomes. The encoder functions are indexed in increasing order so that the encoder is the best when its fitness function is equal to the maximum value and when the BER of the encoder is equal to the minimum value. Whenever the population changes, it is reindexed.

The genetic algorithm was run to optimize the encoder with three variables M = {G, T, E}, the population N = 15; the mutation rate Γ = 0.03, and the culling period Tc = M. The number of mutants δ per child was variable.

Initially, δ = 0. If after Tc generations a new best encoder has not been identified, then δ is incremented up to a maximum of 30. On the other hand, if a new encoder has been identified within Tc generations, then δ is decremented.

The final value of d free (f(x)) is shown in Tables 3 and 4, the optimal generation which identified the best encoder is listed in Table 6.

Table 5 indicates the d free values obtained by Ungerboeck and with the genetic algorithm, indicating the number of generations required by the genetic algorithm to converge. As can be seen, the genetic algorithm with three memories has found the same optimal value for d free as was found by Ungerboeck, however with 4 and 5 memories the optimal values obtained were different but had the same performances as shown in Fig. 11. The best encoder found by the genetic algorithm for 8 states, 16 states and 32 states and the maximum value of d free after going through different generations are shown in Table 6. Finally, Table 7 shows the variation of values from average to best fitness values.

TABLE III. Initial Population

CH Nm	Initial Population	X Value	Fitness Value f(x)	Selection Pro
1	00110001001001000	25160	2.590	0,0637
2	00110001001001100	25164	2.590	0,0637
3	00110001001011000	25176	2.590	0,0637
4	00110001001100000	25184	2.590	0,0637
5	00110001001100100	25188	2.590	0,0637
6	00110001001110000	25200	2.590	0,0637
7	00110001001101000	25192	2.590	0,0637
8	00110001001101100	25196	2.590	0,0637
9	00110001001111000	25208	2.590	0,0637
10	00110001010010100	25236	2.590	0,0637
11	00110001010001100	25228	2.590	0,0637
12	00110001010011000	25240	2.590	0,0637
13	00110001010000100	25220	3.180	0,0782
14	00110001010010000	25232	3.180	0,0782
15	00110001011000100	25284	3.180	0,0782
		Sum	40.62	
		Average	2.708	
		Max	3.18	

The chromosomes that will reproduce are selected based on their fitness values, using the following probability:

$$P(\text{chromosome I reproduces}) = \frac{f(x_i)}{\sum_{k=1}^{15} f(x_k)} \quad (3)$$

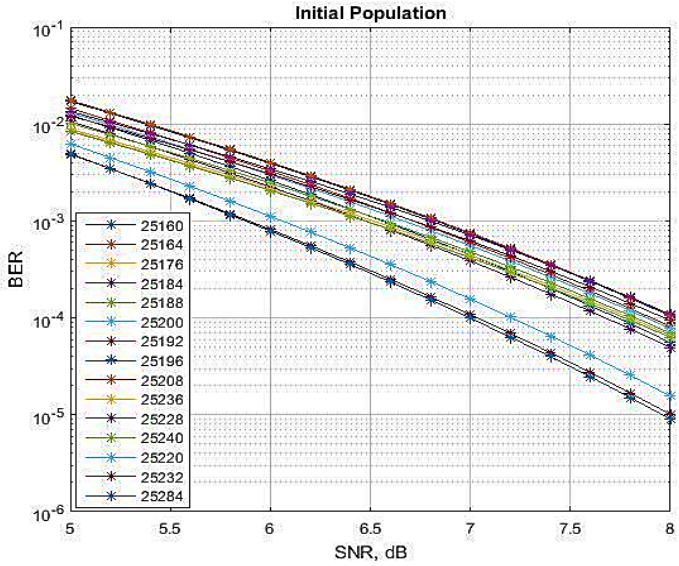


Fig. 6. Comparison between 15 chromosomes of the initial population.

- The simulation illustrated in Fig. 6 shows the comparison between 15 chromosomes of the initial population where the best chromosome corresponds to [00110001011000100] with a d free max value equal to 3.18 and the minimum value for BER (Fig. 7).

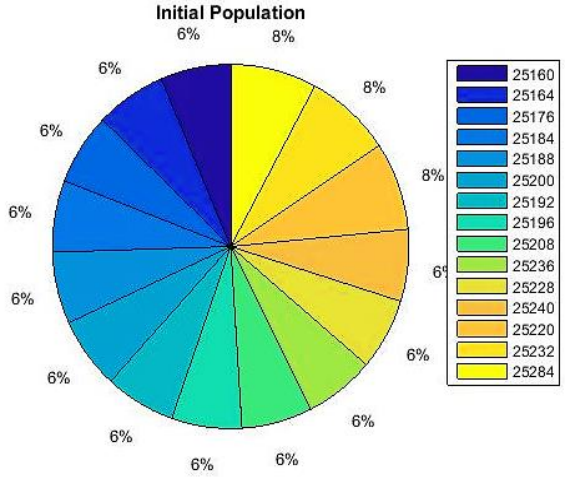


Fig. 7. Presentation of the initial population as a wheel of 15 chromosomes.

TABLE IV. REPRODUCTION AND SECOND GENERATION

CH Nb	New Population	X Value	Fitness Value f(x)	Selection Probability
16	00110001011001000	25288	2.590	0,0503499
17	00110001001100100	25188	2.590	0,0503499
18	00110001010110000	25264	2.590	0,0503499
19	00110001010110100	25268	2.590	0,0503499
20	00110001011100000	25312	2.590	0,0503499
21	00110001011100100	25316	2.590	0,0503499
22	00110001011010000	25296	3.180	0,0618195
23	00110001010011100	25244	3.180	0,0618195
24	00110001011011000	25304	3.180	0,0618195
25	00110001000101100	25132	4.00	0,0777604
26	00110001000111000	25144	4.00	0,0777604
27	00110001000011000	25112	4.590	0,0892301
28	00110001000011100	25116	4.590	0,0892301
29	00110001000100100	25124	4.590	0,0892301
30	001100010001001100	25252	4.590	0,0892301
		Sum	51,44	
		Average	3,429333333	
		Max	4.59	

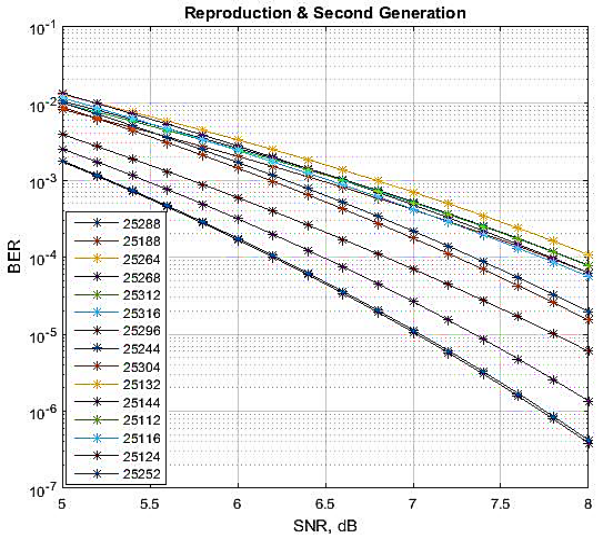


Fig. 8. Comparison between 15 chromosomes of the second generation.

- Now the second generation is tested by the fitness function, and the cycle is repeated. The simulation illustrated in Fig. 8 shows a comparison between 15 chromosomes of the second generation where the best chromosome corresponds to [00110001000101000] and where the standard representation is:

G = [0 0 1; 1 0 0; 0 1 0], T =[0 0;0 1;1 0]; E = [0 0] with a d free max value equal to 4.59 and the minimum value for BER (Fig. 9).

- Fig. 10 shows the first run of a genetic algorithm maximizing the free distance. The red curve is the highest fitness which corresponds to the second generation and the blue curve corresponds to the first generation where average fitness is equal to 3.18. The best solution is reached which corresponds to the chromosome [00110001000101000] with a d free max value equal to 4.59 and the minimum value for BER.

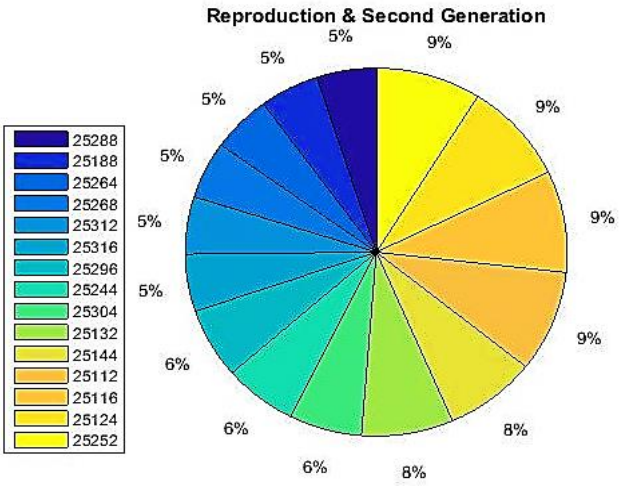


Fig. 9. Presentation of the second population as a wheel of 15 chromosomes.

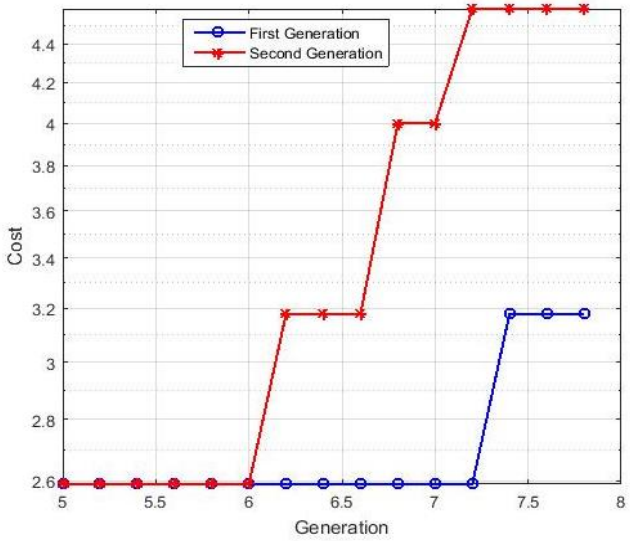


Fig. 10. Comparison between the first and second generations.

TABLE V. VALUE OF D $_{FREE}$ OBTAINED BY UNGERBOECK[1] **[1]** AND THE PROPOSED GENETIC ALGORITHM[2] ALSO LISTED IS THE NUMBER OF GENERATIONS REQUIRED FOR THE GA TO CONVERGE

Number of memories	Modulation	D free from (1)	D free from (2)	Generations
3	PSK	4.586	4.59	20
4	PSK	5.172	5.18	233
5	PSK	5.758	5.77	977

In Fig. 11, the simulation results show that the codes generated by Ungerboeck have the same performance as those generated by genetic algorithm. This is the case after several genetic algorithm code generation iterations conducted until the fitness function value of the chromosome was stabilized and remained stable over many generations at a point where this value converged to the best solution.

TABLE VI. THE BEST CODE GENERATED BY THE GENETIC ALGORITHM

Number of memories	h0	h1	h2	Chromosome
3	$(11)_8$	$(02)_8$	$(04)_8$	001100010 000110 00 → G T E
4	23	16	10	$(0001100101000010\ 00010111\ 00)_2 = (6621276)_{10}$
5	45	01	16	$000011000001001\ 0010000010\ 0110101000\ 00 = (6480733856)_{10}$

TABLE VII. VARIATION OF VALUES FROM AVERAGE TO BEST FITNESS VALUES

Number of memories	Fitness Value f(x)								
Three (03) memories	2,5 9	3.1 8	4	4,5 9					
Four (04) memories	2,5 9	3,1 8	3,7 7	4	4,3 6	4,5 9	5,1 8		
Five (05) memories	2,5 9	3,1 8	3,7 7	4	4,3 6	4,5 9	4,9 5	5,1 8	5.7 7

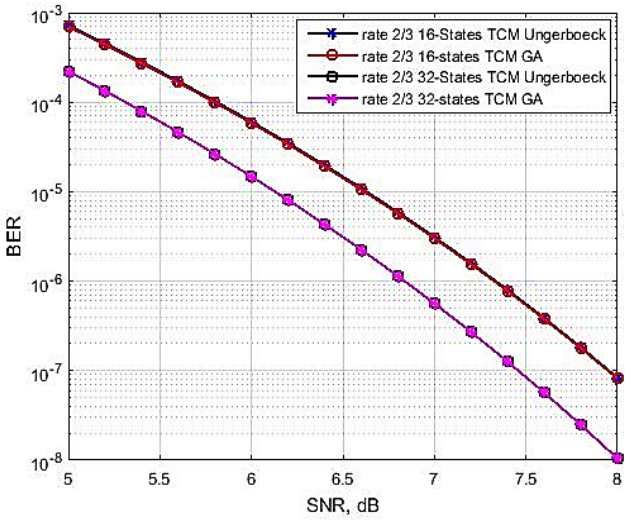


Fig. 11. Comparison between the Ungerboeck codes and those generated by the genetic algorithm in the case of 16 and 32 states.

IV. CONCLUSION

This paper shows improvements which have been obtained by the use of a genetic algorithm for the optimization of TCM schemes, as opposed to other optimization algorithms for this encoder. There are other important details such as the crossover and mutation probabilities, the population size and the iteration number. By applying an appropriate genetic algorithm, a new TCM encoder scheme could be evolved in a multipath channel. These values of the TCM encoder can be adjusted after assessing the algorithm's performance on a few trial runs. Future research attempts to combine genetic algorithms with other optimization algorithms as well as other branches of evolutionary computation, such as neural networks.

REFERENCES

[1] G. Ungerboeck, 'Channel coding with multilevel/phase signals', IEEE Transactions on Information Theory, vol. 28, no. 1, pp. 55-67, 1982.

[2] M. Rouanne, D.J. Costello,"An algorithm for computing the distance spectrum of trellis codes", IEEE J. of SAC, SAC-7(1989)6, 929–940.

[3] Matthew C. Valenti and Raghu Doppalapudi, Don Torrieri, "A Genetic Algorithm for Designing Constellations with Low Error Floors", In IEEE Conference on Information Sciences and Systems, pp. 1155–1160 (March 2008).

[4] Y. Liu, Z. Bie, K. Niu, and Z. He, "Optimization of constellation labeling using genetic algorithm," in IET International Communication Conference on Wireless Mobile and Computing (CCWMC 2009), Dec. 2009, pp. 365– 368.

[5] Confessore G., Di Gennaro M., Ricciardelli S. (2001) A Genetic Algorithm to Design Satellite Constellations for Regional Coverage. In: Fleischmann B., Lasch R., Derigs U., Domschke W., Rieder U. (eds) Operations Research Proceedings. Operations Research Proceedings, vol 2000. Springer, Berlin, Heidelberg.

[6] Tania Savitri,1 Youngjoo Kim,1 Sujang Jo,2 and Hyochoong Bang1. (2017) Satellite Constellation Orbit Design Optimization with Combined Genetic Algorithm and Semianalytical Approach. International Journal of Aerospace Engineering Volume 2017 (2017), Article ID 1235692, 17 pages.

[7] Anit Kumar ENCODING SCHEMES IN GENETIC ALGORITHM (2013) International Journal of Advanced Research in IT and Engineering , ISSN: 2278-6244

[8] Mitchell, M. (2009). Complexity: A Guided Tour. New York: Oxford University Press.

[9] D. Divsalar and F. Pollara, "On the Design of Turbo Codes," JPL TDA Progress Report 42-123, Nov 15, 1995.

[10] S. Friedberg and A. Insel, "Introduction to Linear Algebra with Application," Prentice-Hall, Englewood Cliffs, N. J., 1986.

[11] Mitchell, M. (1995). Genetic Algorithms: An Overview. Complexity, 1(1), 31-39.

[12] Mitchell, M. (1996). An Introduction to Genetic Algorithms. Cambridge: MIT Press.

[13] Rahul Malhotra, Narinder Singh & Yaduvir Singh Design for Optimization of Process Controllers: Computer and Information Science Vol. 4, No. 2; March 2011.