

# Genetic Optimization of LDPC Codes to Improve the Correction of Burst Errors

Jan Broulím<sup>1,2,5</sup>, Alexander Ayriyan<sup>2,5</sup>, and Hovik Grigorian<sup>3,4,5</sup>

<sup>1</sup>University of West Bohemia, Pilsen, Czech Republic

<sup>2</sup>Institute of Technical and Experimental Physics, CTU in Prague, Czech Republic

<sup>3</sup>Yerevan State University, Yerevan, Armenia

<sup>4</sup>A. I. Alikhanian National Laboratory (YerPhI), Yerevan, Armenia

<sup>5</sup>Laboratory of Information Technologies, JINR, Dubna, Russia

## Abstract.

Error correction plays a crucial role when transmitting data from the source to the destination through a noisy channel. It has found many applications in television broadcasting services, data transmission in radiation harsh environment (e. g. space probes or physical experiments) or memory storages influenced by Single Event Effects (SEE). Low Density Parity Check (LDPC) codes provide an important technique to correct these errors.

The parameters of error correction depend both on the decoding algorithm and on the LDPC code given by the parity-check matrix. Therefore, a particular design of the parity-check matrix is necessary. Moreover, with the development of high performance computing, the application of genetic optimization algorithms to design the parity-check matrices has been enabled.

In this article, we present the application of the genetic optimization algorithm to produce error correcting codes with special properties, especially the burst types of errors. The results show the bounds of correction capabilities for various code lengths and various redundancies of LDPC codes. This is particularly useful when designing systems under the influence of noise combined with the application of the error correction codes.

## 1 Introduction

Error correction plays a crucial role in the data transmission chain from a source to a destination through a noisy channel. The noisy channel represents the addition of errors to the transmitted data stream. Such errors are usually caused by an unexpected interference, electronics failure, etc. Radiation harsh environment is one of the most common applications of error correction due to the Single Event Effects occurring in electronics system. The error correction is also necessary for obtaining correct results in physical experiments.

The Low Density Parity-Check (LDPC) codes [1], [2] are linear block codes defined by a low density parity-check matrix. The codeword is obtained by  $\mathbf{c} = \mathbf{m}\mathbf{G}$ , where  $\mathbf{m}$  is an information vector (message),  $\mathbf{c}$  is a codeword, and  $\mathbf{G}$  is the generator matrix associated with the code. The condition  $\mathbf{c}\mathbf{H}^T = \mathbf{0}$  is satisfied if and only if  $\mathbf{c}$  is a codeword.

The correction performance is given either by the structure of  $\mathbf{H}$  and decoding algorithm used. A particular design of the parity-check matrix is necessary [1]. In the next sections, we present selected results from the design of the parity-check matrix  $\mathbf{H}$  to correct the burst type of errors.

In this work, we show the results obtained by the application of the optimization algorithm proposed in [3]. The optimization algorithm is based on producing pseudo-random parity-check matrices using so called mutations. SP algorithm for LDPC decoding was chosen due to its generality without any simplifications [4].

## 2 Modeling of multiple burst errors

Usually not only one single bit in the data stream is affected by the failure, but also some of its neighbors. Suppose the communication from the source to the destination in terms of binary vector transmission. The codeword  $\mathbf{c}$  is transmitted through a noisy channel and affected by the error vector  $\mathbf{e}$ . Then, the receiver gets a corrupted codeword  $\mathbf{c}' = \mathbf{c} + \mathbf{e}$ .

The **burst error** is a type of error when the vector  $\mathbf{e}$  contains a cluster of 1's surrounded by all 0's.

The **multiple burst** error is an error when the vector  $\mathbf{e}$  contains more than one such cluster.

The multiple bursts are very common errors in real environment. Here we present a method how to design parity-check matrices associated with LDPC codes in order to provide the best correction capabilities for this type of errors. We show that the genetic optimization is a highly convenient method to this purpose and show a number of selected results.

The burst error pattern  $E_P$ , used as an optimization parameter, is given by

$$E_P = (x_1, x_2, \dots, x_k). \quad (1)$$

Here  $x_i$  denote the lengths of error bursts in the codeword and  $|E_P|$  is the number of burst errors. The total number of corrupted bits is then  $\sum_{i=1}^k E_P(i)$ .

For an LDPC decoder [4], we define the initial probabilities  $p_j(a) = p(c_j = a|y_j)$  as the amount of belief that the  $j$ -th symbol is equal to  $a$ , where  $y_j$  is the information received from the channel. Considering the vector  $\mathbf{e}$ , we define the average probability of the error  $p^e(a) = N_e/n$ , where  $N_e$  is the number of 1's in the vector  $\mathbf{e}$  and  $n$  is the length of codeword. Then,

$$p_j(a) = \begin{cases} p^e & \text{if } c_j = 0, \\ 1 - p^e & \text{if } c_j = 1. \end{cases} \quad (2)$$

This model is used during the genetic optimization for the calculation of the fitness function.

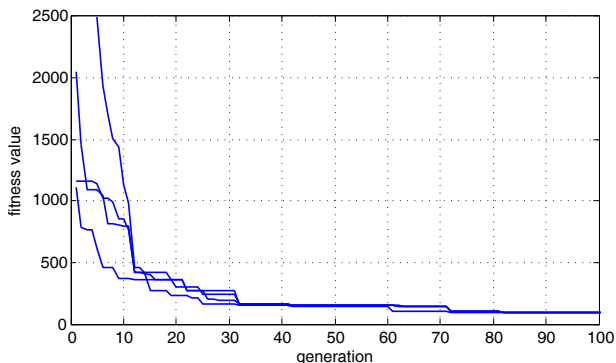
## 3 Application of the genetic optimization algorithm

The main optimization task consists of several steps: Producing the initial population, evaluation using the fitness function, selection of the best results, and producing the next generation. After producing the next generation, the algorithm continues the evaluation and selection in the loop. It terminates after a given time period or a number of generations. The next generations are produced using mutations of the parity-check matrix  $\mathbf{H}$ , described in [3]. The Sum-Product (SP) algorithm is used for decoding [4]. For code evaluation, we use the fitness function defined below.

The fitness function checks all combinations of error bursts given by burst lengths as elements of the tuple  $E_P$  and calculates the number of corrupted codewords after the correction. The fitness function  $f$  is expressed as the total number of corrupted codeword after decoding for a given burst error pattern. This type of evaluation is deterministic.

## 4 Results

In this section, we present selected results from the genetic optimization. The correction performance for double burst error correction measured for different codes and different burst lengths are given in Tables 1–2. The code rate  $R$  is equal to  $R = k/n$ , where  $k$  is the information message length and  $n$  is the codeword length. The optimization tasks were held for 100 generations and the sharing period [3] was 10 generations. Fig. 1 plots the dependency of best fitness values on the generation number for selected optimization task. The tasks were run on HybriLIT and Govorun supercomputing clusters [7].



**Figure 1.** Sample optimization tasks. The LDPC code (128,72) has been selected for the optimization. The convergence of the best fitness value (the best value in the population produced in the particular generation) to the optimal solution can be seen.

**Table 1.** Correcting capabilities (%) measured by the ratio of frames which are able to be corrected. The burst length corresponds to the definition given in Eq. 1

$n$	$k$	$R$	tuple of burst lengths												
			(1,1)	(2,1)	(3,1)	(3,2)	(4,2)	(4,3)	(5,3)	(5,4)	(6,4)	(6,5)			
128	96	0.75	98.8	90.1	55.8										
128	72	0.5625				98.8	96.8	93.4	81.2	52.4					
128	64	0.5				100	99.8	99.2	96.2	89.7	77.5	61.9			
			(5,3)	(5,4)	(6,4)	(6,5)	(7,5)	(7,6)	(8,6)	(8,7)	(9,7)	(9,8)			
128	56	0.4375	99.7	99.4	98.0	93.9	84.5	71.0							
128	48	0.375			100.0	99.5	98.6	94.8	87.3	74.7	55.1	40.0			
			(7,5)	(7,6)	(8,6)	(8,7)	(9,7)	(9,8)	(10,8)	(10,9)	(11,9)	(11,10)			
128	40	0.3125	100.0	99.6	98.2	97.1	93.8	89.6	78.0	60.2					
128	32	0.25		100	100	99.9	99.8	99.0	98.2	95.1	88.9	80.1			

## 5 Conclusions

In this article, we presented the application of genetic optimization algorithms for improving the performance correction capability of LDPC codes for a specific type of errors, namely error bursts. Such errors are very common in a data transmission influenced by any type of interference. The results show the bounds of correction capabilities for various code lengths and various redundancies.

**Table 2.** Correcting capabilities (%) measured by the ratio of frames which are able to be corrected

<i>n</i>	<i>k</i>	<i>R</i>	tuple of burst lengths							
			(1,1)	(2,1)	(3,1)	(3,2)	(4,2)	(4,3)	(5,3)	(5,4)
256	192	0.75	99.7	97.9	93.6	84.5	59.8	29.4	6.3	
256	176	0.6875	96.6	93.3	88.6	75.6	51.5	23.4	10.7	
256	144	0.5625	99.8	99.7	99.5	97.2	94.3	91.6	85.2	73.3
256	128	0.5	100	100	99.8	99.6	98.9	97.3	95.8	92.8
256	96	0.375	100	99.9	99.9	99.7	99.4	99.1		

A very important group of the codes are codes with 50% redundancy. From the tested cases, the code (128,64) is able to correct all bursts of length (3,2), the code (256,128) is able to correct all bursts of length (7,5), and the code (512,256) is able to correct all bursts of length (9,7). When longer bursts occur, the error correction cannot be guaranteed. The results are particularly useful in the generation of new codes for specific applications and to know the properties of electronic systems under the influence of noise.

### Acknowledgement

This work was supported by the grant no. 05-6-1119-2014/2019 of the Plenipotentiary of the Czech government and by the project SGS-2018-001 “Research and development of modern methods for electronic and communication systems in scientific and engineering applications”.

### References

- [1] G. Liva, S. Song, L. Lan, Y. Zhang, S. Lin, W.E. Ryan, *Journal of Communications Software and Systems* **2**, 3, 191–211 (2006)
- [2] D. Mackay, D. MacKay’s Gallager code resources.  
Available: <http://www.inference.org.uk/mackay/codes/> Accessed: 15/11/2019
- [3] J. Broulim, S. Davarzani, V. Georgiev, J. Zich, *Telecommunications Forum* **24**, 250–253 (2016)
- [4] N. Wiberg, *Codes and Decoding on General Graphs* (PhD thesis, Dept. of Electrical Engineering, Lionköping, Sweden, 1996. Lionköping studies in Science and Technology, Dissertation No. 440, 1996). Available:  
[https://www.researchgate.net/publication/2408031\\_Codes\\_and\\_Decoding\\_on\\_General\\_Graphs](https://www.researchgate.net/publication/2408031_Codes_and_Decoding_on_General_Graphs). Accessed: 15/11/2019
- [5] Y. Yu, Z. Jia, W. Tao, S. Dong, *Global Conference on Consumer Electronics* **5**, 607–607 (2016)
- [6] X. Qin, C. Yang, Z. Zheng, Z. Wang, *Photonics Technology Letters*, **31**, 1401–1404 (2019)
- [7] Gh. Adam, M. Bashashin, D. Belyakov, M. Kirakosyan, M. Matveev, D. Podgainy, T. Sapozhnikova, O. Streltsova, Sh. Torosyan, M. Vařa, L. Vařová, A. Vorontsov, T. Zaikina, E. Zemlyanaya, M. Zuev, *CEUR Workshop Proceedings* **2267**, 638–644 (2018)
- [8] L. Dudacek, I. Vertat, *Telecommunications Forum* **24**, 258–261 (2016)
- [9] J. Broulim, A. Ayriyan, H. Grigorian, Application note: LDPC (128,64). Available:  
[http://home.zcu.cz/~broulim/LDPC/ldpc128\\_burst\\_optimized.pdf](http://home.zcu.cz/~broulim/LDPC/ldpc128_burst_optimized.pdf). Accessed: 15/11/2019.