

Analysis of genetic programming's parameters for 2D Wavelet Sign Coding

J. M. Navarro¹, O. López-Granado¹ and M. P. Malumbres¹

Abstract -- In this paper we perform an analysis of the optimal parameters values for a genetic algorithm. This genetic algorithm will predict the sign of wavelet coefficients based on the sign information of an Intra-band neighborhood in the context of image encoding

Keywords -- sign coding, discrete wavelet transforms, image coding, genetic algorithms

I. INTRODUCTION

IN this work we are looking for the optimal parameters for a wavelet image compressor sign predictor based on a genetic algorithm. This kind of image compressor is based in the use of a mathematical transform called Discrete Wavelet Transform (DWT). Wavelet transforms have proved to be very powerful tools for image compression, since many state-of-the-art image codecs, including the JPEG2000 standard [1], employ DWT into their algorithms. One advantage of the wavelet transform is the provision of both frequency and spatial localization of image energy. The image energy is compacted into a small fraction of the transform coefficients and compression can be achieved by coding these coefficients. The energy of a wavelet transform coefficient is restricted to non-negative real numbers, but the coefficients themselves are not, and they are defined by both a magnitude and a sign. Shapiro stated in [2] that a transform coefficient is equally likely to be positive or negative and thus one bit should be used to encode the sign. In recent years, several authors have begun to use context modeling for wavelet sign coding [3][1][4], showing that despite the equiprobability of wavelet sign values, some sign correlation can be found among wavelet coefficients, resulting in overall compression ratio improvements. In a previous work [5] we have observed that the sign of a wavelet coefficient may be strongly correlated with the sign of some neighbor coefficients.

On the other hand, Genetic algorithms (GA) were first introduced by Holland in [6] and they are nowadays well known techniques for finding nearly optimal solutions of very large problems and also, they have been used in image processing [7][8]. In a genetic algorithm, the evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated by means of a cost function that determines the optimal degree we are looking for (i.e. compression rate). Multiple individuals are

stochastically selected from the current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. In this paper, our aim is to analyze every possible simulation of the genetic algorithm and find its optimal parameters.

The remainder of the paper is organized as follows: in Section 2 we define the optimization problem and propose the genetic algorithm that matches the problem definition. Section 3, we analyze the simulation results. Finally, in Section 4 we draw some conclusions.

II. WAVELET SIGN PREDICTION: PROBLEM STATEMENT

To estimate sign correlation in a practical way, we have applied a 6-level Dyadic Wavelet Transform decomposition of the source images. As the sign neighborhood correlation depends on the subband type (HL, LH, HH), we have used different neighbors depending on the subband type and the number of neighbors we have utilized, as we tested the algorithm with 4 neighbors. So, for the HL subband, the neighbors used are N (North), NN (North-North), W (West) and WW (West-West). Taking into account symmetry, for the LH subband, those neighbors are W, WW, N and NN. For the HH subband they are W, N, NW (North-West) and NNWW (North-North West-West), exploiting the correlation along and across the diagonal edges. This leads us to a maximum of 3⁴ Neighbor Sing Patterns (NSPs) for each subband type. After running the genetic algorithm for each subband type, we obtain the sign prediction table that contains the sign predictions for every pattern. So, when coding the sign of a wavelet coefficient in a particular subband, first we will get the sign value of the corresponding neighbor set in order to form the actual pattern. Then, we will compare the sign prediction for that pattern with the current coefficient sign and we encode whether it is correct or not. Thus, the performance of a binary entropy encoder will depend on the behavior of our sign predictor, the higher the success prediction ratio the higher the compression rate.

A. Genetic Algorithm Definition

The genetic algorithm created follows the classic structure for this type of algorithms: population initialization, population evaluation and new population generation through the classification of individuals using

¹ Physics and Computer Architecture Department. Miguel Hernández University. e-mail: jmnnavarro90@gmail.com, {otoniel, mels}@umh.es

a fitness function and the use of crossover and mutation operators to create new individuals. Then we need to create a population (universe) of individuals that during the evolution process will improve their goodness in base to a fitness function that will determine their quality. The genetic information of one individual is the set of sign prediction values of every NSP[k]. For our purposes we will define the fitness function in such a way that its result indicates the sign prediction performance of one individual for the set of images. In other words, the fitness function will estimate the compression rate of the sign prediction encoding that would be achieved if the prediction table defined by this individual is used to encode the image. Finally, single point crossover is used in all variations where the locus point to split the parent gene is randomly selected. On the other hand, the mutation policy inverts the prediction value of a randomly chosen gene. Also, two best individuals survive to the next generation and they cannot be modified by the mutation operator.

Each individual is represented by a binary vector where its elements represent a combination of signs from a predefined neighborhood set of coefficients, and the stored values into these elements correspond to the sign prediction for the coefficient (binary value). The size of this vector depends on the number of neighbors that conforms the neighborhood. The greater the number of neighbors considered, the greater then number of sign combinations, namely 3^n being n the number of neighbors, since the possible sign values of neighbor wavelet coefficients are ternary values (positive, negative or non-significant).

As our aim is to find which values in the execution of the algorithm provide us the optimal fitness, we have run a battery of tests, namely, we have used this sets of values for the algorithm parameters:

- Mutation probability: 0.01, 0.025, 0.05, 0.075, 0.1, 0.25, 0.5, 0.75, 1.
- Population number: 300, 280, 260, 250, 220, 200, 180, 160, 150, 140, 120, 100, 80, 60, 50, 40, 20, 10.
- Number of rounds: 100, 200, 300, 400, 500, 1000, 3000, 5000, 10000.

Due to time constraints, all the simulations were done only on one image, the first one of the Kodak image set [9].

The program implementation of the algorithm, written in pseudocode, is as follows:

```

Individual Structure/
sign[NSP]; //Prediction array for each neighbor sign pattern
//combination
fitness; //indicates the goodness of the individual
}Individual universe[NUM-POPULATION]; //Individual array

function SignPrediction (SubbandType, ImageFiles, mutation
Probability)
//Initialization phase: sign[NSPs]= random(POSITIVE/NEGATIVE)
Initialize (universe, NUM-POPULATION, NSP); //we evaluate each
//individual of the universe. For each //image in ImageFiles
EvaluateFitness (SubbandType, ImageFiles, universe);
for i=0 to NUM-ITERATIONS
//Select the best two individuals from universe for survival.
best = SelectBestIndividuals (2);
//Crossover
crossPoint=random (NSP);

```

```

//randomly selects a father and a mother to mix its gens
SelectFatherAndMother (random (NUM-POLUTATION));
universe = MergeFatherAndMother (crossPoint);
Mutation (universe, mutation Probability);
universe = universe + best;
EvaluateFitness (SubbandType, ImageFiles, universe);

end
//Finally get the best individual.
best = SelectBestIndividuals (1);
end of function

```

III. PERFORMANCE EVALUATION

A. Convergence Evaluation

The way we are going to determine the GA optimal parameters in this chapter is by studying its convergence based on them. So, for each subband, we will analyze the number of simulations that don't converge to the maximum value attainable within that simulation framework and, thus, we will obtain a broad view of the parameters' adequate values.

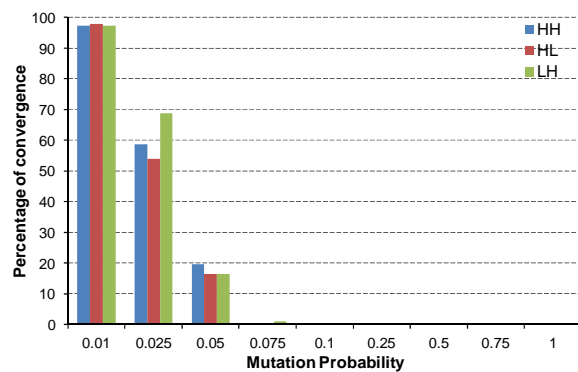


Fig. 1. Percentage of convergence based on mutation probability

As we can see in Fig. 1, using a mutation probability value higher than 0.05, the GA does not converge even for a high number of rounds. Based on this, we can deduce the optimum value to use would be 0.01, in terms of convergence.

Regarding population amount parameter, the results on Fig. 2 show that population number does not have an important effect in the convergence of the algorithm, it probably does affect the overall fitness obtained, which we will analyze further in this article.

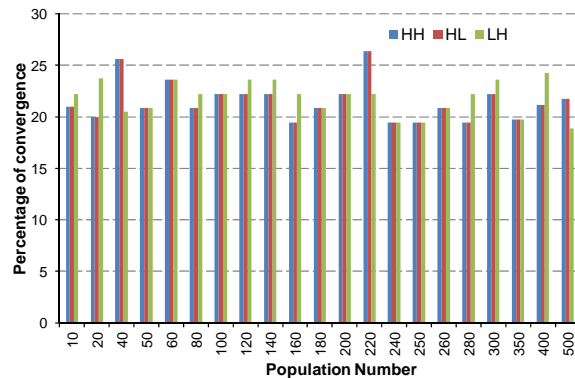


Fig. 2. Percentage of convergence based on population number

Lastly, we check the effect of the number of rounds in the convergence of the GA. Logically, as we increase the number of rounds the amount of simulations that

converge increase, although we can assume the low percentages are due to the number of jobs done with a high mutation probability because as we have already seen, it makes the GA oscillate around the optimum solution (see Fig. 3).

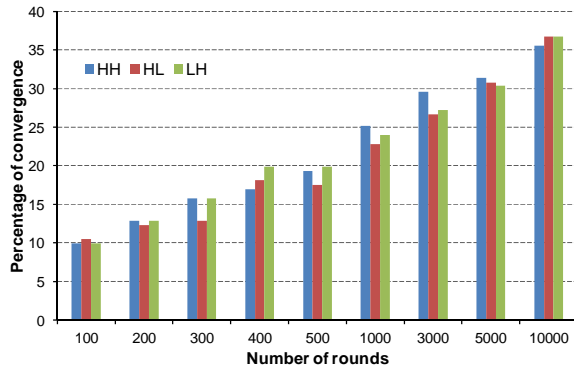


Fig. 3. Percentage of convergence based on number of rounds

B. Fitness Evaluation

Now that we have studied the convergence properties of the GA based on its input parameters, we can analyze the overall fitness obtained as a function of them. We will do as in the previous section; we will separate the results in three charts, with three different data sets in each one, one for each subband type. Thus, we start with the mutation probability. As we can observe on Fig. 4, as the mutation probability decreases, so does the fitness obtained, with the exception of the mutation probability value of 1, which suffers a slight increase. This is due to a greater number of NSPs changing as the mutation probability increases, making the fitness value to oscillate not allowing the convergence.

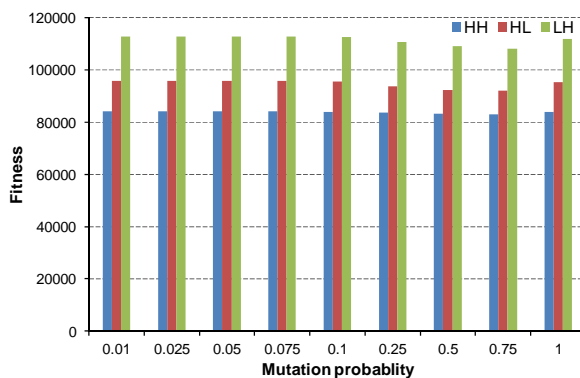


Fig. 4. Fitness based on mutation probability

In Fig. 5 we show the fitness value as a function of the population number. We can conclude that the population number does not affect the maximum fitness value obtained.

Finally, in Fig. 6 we find that the fitness value of the GA is not affected by the number of rounds, as it reaches its maximum for a hundred rounds in certain conditions.

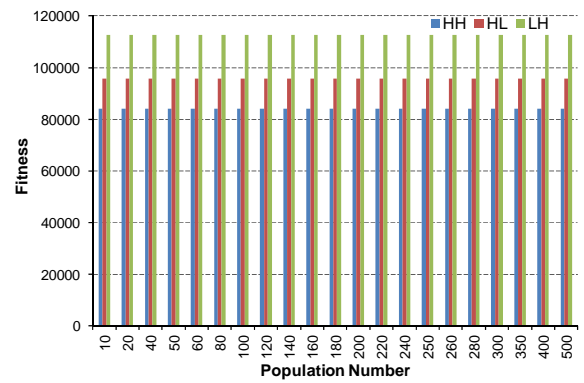


Fig. 5. Fitness based on population number

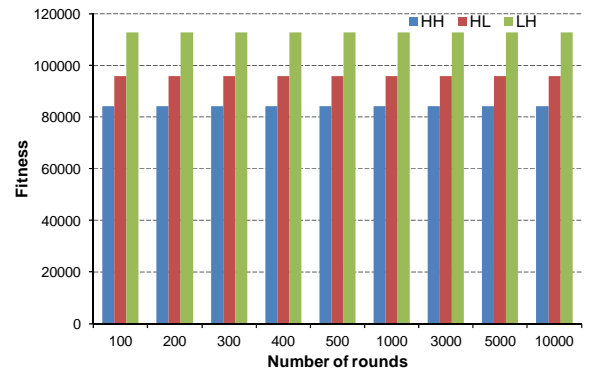


Fig. 6. Fitness based on number of rounds

Now, having all the results we can easily see that the defining parameter for convergence and fitness is the mutation probability. We have checked, also, that the single value that produces the highest fitness and convergence ratio is 0.01, so, in the next section we are going to center in the results of the simulations run with that probability value and see if we can fine tune the GA even more.

IV. TUNING THE GENETIC ALGORITHM

In fact, for this image (as we must remember every result we have drawn only applies to the image we have analyzed), using 0.01 as the mutation probability almost always produces the optimum value attainable. As such, we do not even need to plot the results, we can just specify which sets of parameters do not produce the optimum fitness value or do not converge. Tables I, II, and III show these parameters values for subbands HH, HL and LH respectively.

TABLE I
HH SUBBAND SETS OF PARAMETERS THAT DO NOT CONVERGE OR PRODUCE OPTIMUM FITNESS

Population	Number of rounds	Fitness	Maximum Fitness
80	100	84072	84074
100	100	84072	84074
10	200	84054	84074
20	100	84036	84074
10	100	83922	84074

TABLE II

HL SUBBAND SETS OF PARAMETERS THAT DO NOT CONVERGE OR PRODUCE OPTIMUM FITNESS

Population	Number of rounds	Fitness	Maximum Fitness
20	100	95656	95684
10	200	95628	95684

TABLE III

LH SUBBAND SETS OF PARAMETERS THAT DO NOT CONVERGE OR PRODUCE OPTIMUM FITNESS

Population	Number of rounds	Fitness	Maximum Fitness
10	300	112762	112770
40	100	112760	112770
50	100	112738	112770
10	100	112680	112770
20	100	112662	112770

A. Neighborhood Increment

To further expand our analysis, now we run the GA with the optimal parameters we found before (0.01 as mutation probability, 100 as population number and 100 as number of rounds) but, this time, we change the number of neighbors to be analyzed. We are going to use the neighborhood showed in Table IV.

TABLE IV
NEIGHBORS SETS USED

Number of neighbors	Subband type	Neighbors used
3	HH	N, W, NW
	HL	N, NN, W
	LH	W, WW, N
4	HH	N, W, NW, NNWW
	HL	W, WW, N, NN
	LH	N, NN, W, WW
5	HH	W, N, NW, NNWW, NNNWWW
	HL	N, NN, NNN, W, WW
	LH	W, WW, WWW, N, NN

In Fig. 7, we compare the results, in normalized fitness (see Eq. 1) for the HH, HL and LH subbands based on the different neighbors sets.

$$\frac{\sum_{i=0, j=0}^{N, M} \sum_{k=1}^3 \overline{SC_{i,j}^{NSP}[k] \oplus SC_{i,j}}}{M \times N} \forall image \quad (1)$$

Where N, M are the image dimensions, $\overline{SC_{i,j}[k]}$ is the sign prediction for NSP (k) and $\overline{SC_{i,j}}$ is the sign of wavelet coefficient $C_{i,j}$. The division by N x M is performed to normalize the fitness function because the different evaluated images could be of different sizes.

As shown in Fig. 7, for HH and LH subbands, we can clearly see that fitness value increases as the number of neighbors increase. Even more, with five neighbors, the fitness obtained suffers a significantly increase, being

this option the best choice for these subband types. Contrary to HH and LH subbands, and for this image, the best choice for HL subband is to use only three neighbors. This effect is mainly due to the low amount of vertical frequencies in this image, and so, there are less significant coefficients to exploit the vertical correlation between them.

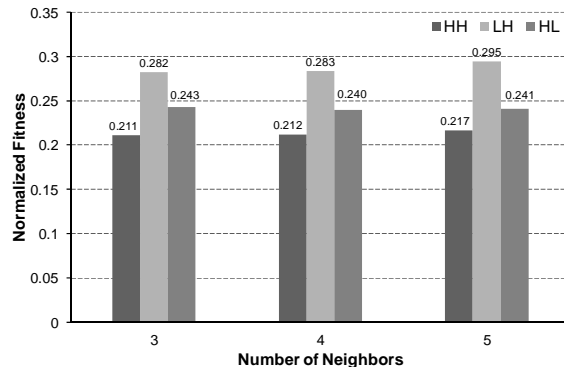


Fig. 7. Normalized fitness for HH,HL and LH subbands based on number of neighbors

V. CONCLUSIONS AND FURTHER WORKS

As we have seen and already stated, this was a limited analysis, due to time and processing power constraints, so this results should be taken as a first approximation to a broad analysis of the GA as an effective sign predictor. The single and most important conclusion we can draw from this results is that the defining parameter in the convergence and fitness of the GA is the mutation probability. And so, using 0.01 as mutation probability and avoiding the parameter sets we have specified in section 3.3, we can assure the best results for image 01 of the Kodak image set are obtained.

Furthermore, we have determined that, based on the type of subband to analyze, there is an optimum neighbors set to use, namely, three neighbors for the HL subband and five for the HH and LH subbands.

As future work, we will continue fine tuning the algorithm and, with that objective in mind, the next step is to apply the same type of analysis to a broader set of images from the Kodak image set, with the aim of producing general results which can be applied to an image codec, which is the ulterior objective of the development of the GA.

VI. ACKNOWLEDGMENTS

Thanks to Spanish Ministry of Education and Science under grants TIN2011-27543-C03-03 and TEC2010-11776-E for funding.

VII. REFERENCES

- [1] Taubman, D.: *High performance scalable image compression with EBCOT*. IEEE Transactions on Image Processing 9(7) (July 2000) 1158-1170
- [2] Shapiro, J.: *A fast technique for identifying zerotrees in the EZW algorithm*. Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing 3 (1996) 1455-1458
- [3] Wu, X.: *High-order context modeling and embedded conditional entropy coding of wavelet coefficients for image compression*. In: Proc. of 31st Asilomar Conf. on Signals, Systems, and Computers. (1997) 1378-1382

- [4] Deever, A., Hemami, S.S.: *What's your sign?: Efficient sign coding for embedded wavelet image coding*. In: Proc. IEEE Data Compression Conf., Snowbird, UT. (2000) 273–282
- [5] Lopez, O., Martinez, M., Pinol, P., Malumbres, M., Oliver, J.: *E-LTW: An enhanced ltw encoder with sign coding and precise rate control*. In: Image Processing (ICIP), 2009 16th IEEE International Conference on. (nov. 2009) 2821 – 2824
- [6] Holland, J. In: *Adaption in Natural and Artificial Systems*. University of Michigan Press (1975)
- [7] Chabrier, S., Rosenberger, C., Emile, B., , Laurent, H.: *Optimization-based image segmentation by genetic algorithms*. EURASIP Journal on Image and Video Processing 2008 (2008) 1–10
- [8] Anam, S., Islam, M.S., Kashem, M., Islam, M., Islam, M., Islam, M.: *Face recognition using genetic algorithm and back propagation neural network*. In: International MultiConference of Engineers and Computer Scientists, Hong Kong (2009)
- [9] Center for Image Processing Research -Electrical, Computer, and Systems Engineering Department - Rensselaer Polytechnic Institute
<http://www.cipr.rpi.edu/resource/stills/kodak.html>