



ELSEVIER

Pattern Recognition Letters 18 (1997) 311–318

Pattern Recognition
Letters

Genetic annealing search for index assignment in vector quantization

Tomasz Ostrowski^{a,b,*}, Vesa T. Ruoppila^{c,1}

^a Tampere University of Technology, Digital Media Institute, P.O. Box 553, FIN-33101 Tampere, Finland

^b Technical University of Szczecin, Institute of Electronics and Computer Science, ul. 26 Kwietnia 10, PL-71126 Szczecin, Poland

^c Nokia Research Center, P.O. Box 100, FIN-33721 Tampere, Finland

Received 3 September 1996; revised 1 April 1997

Abstract

A Genetic Annealing Algorithm (GAA) has been applied to an Index Assignment (IA) problem in Vector Quantization (VQ). The proposed combination of annealing, selection and recombination mechanisms is new and possesses an advantage with regard to an implementation on a single processor machine for the reason of satisfactory performance when using a reasonably small population. The assessment of the proposed method by using a vector quantization of Line Spectrum Frequencies (LSF) of a speech coder, the experiments with the vector quantizers trained on the Gaussian data as well as the tests on the Quadratic Assignment Problem (QAP) benchmarks are presented. © 1997 Elsevier Science B.V.

Keywords: Genetic annealing; Vector quantization; Index assignment; Quadratic assignment problem

1. Introduction

In vector quantization (Gersho and Gray, 1992) a source vector is encoded by assigning to it one codevector from a set of the codevectors stored in the codebook. An index that identifies the selected codevector is transmitted across a communication channel in a binary form. At the receiving end the index is decoded with the corresponding codevector from the codebook. However, bit errors occurring in the channel may result in an incorrectly received

index. Thus the resulting codevector may no longer be the best representation of the source vector and the quantization distortion may increase due to channel noise.

An index assignment problem, known also as a non-redundant channel coding, can be stated as follows. Given the codebook of 2^n codevectors, find the optimal assignment of a unique n -bit index to each of the codevectors to minimize the channel distortion. In general any change in the assignment of indices to the codevectors can affect the quantization distortion resulting from channel errors. An assignment of indices to the codevectors is equivalent to a permutation of a given codebook. The total number of permutations grows as a factorial $2^n!$ of

* Corresponding author. E-mail: ostrowsk@arcadia.tuniv.szczecin.pl.

¹ E-mail: ruoppila@research.nokia.com.

the size of the codebook, so an exhaustive search over the whole codebook permutation set is unfeasible for $n > 3$. However, some suboptimal approaches to the problem have been presented.

For instance, Zeger and Gersho (1990) introduced a binary switching algorithm where a pair of vectors is swapped in an iterative fashion, while Farvardin (1990) employed a simulated annealing strategy. Potter and Chiang (1995) proposed a method employing a minimax criterion. It has also been demonstrated experimentally that a properly constructed splitting LBG algorithm (Linde et al., 1980) results in a better index assignment than a random indexing of the codebook.

Knagenhjelm and Agrell (1996) have formulated the index assignment problem by using the Hadamard transform and introducing the concept of linearity of the codebook. They developed the Linearity Increasing Swap Algorithm (LISA) as a tool for index assignment, which employs the codebook linearity as the search criterion. However, it should be pointed out that the search for the solution optimal in terms of codebook linearity is correlated but not equivalent to the search for an optimal index assignment. Moreover, their approach is restricted by a Maximum Entropy Encoder (MEE) assumption, i.e. assuming all codevectors to be used with equal probability, and by employing the distance measure between the codevectors determined by the squared Euclidean norm.

In this paper we propose a genetic annealing approach to an index assignment problem and present the results of the GAA assessment by using an LSF quantization of a speech coder and the vector quantizers trained on the Gaussian data. Additionally we test the GAA performance on the QAP, which was first formulated by Koopmans and Beckmann (1957).

2. Problem formulation

We assume the following channel distortion measure (Farvardin, 1990):

$$E = \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} P(c_i) P(b(c_j) | b(c_i)) d(c_i, c_j), \quad (1)$$

where E denotes expected channel distortion, $P(c_i)$ is the a priori probability of using a codevector c_i , $d(c_i, c_j)$ is a distance measure between two codevectors, M is the number of codevectors in the codebook, $b(c_i)$ is a one-to-one mapping that assigns a binary index to the codevector c_i , and $P(b(c_j) | b(c_i))$ is the conditional probability that the index $b(c_j)$ of the codevector c_j is received given that the index $b(c_i)$ is transmitted. If the channel used for the transmission of the indices is a binary symmetric channel with a Bit Error Probability (BEP) e_k for the k th bit of an index, then the conditional probability $P(b(c_j) | b(c_i))$ for the indices $b(c_i) = p$ and $b(c_j) = q$ can be evaluated as follows (Farvardin, 1990):

$$P(q | p) = \prod_{k=1}^n (1 - e_k)^{(1 - l_k(p, q))} e_k^{l_k(p, q)}, \quad (2)$$

where n is the index length, $l_k(p, q) = 1$ if the binary indices p and q differ in the k th position and $l_k(p, q) = 0$, otherwise. In our approach the expected channel distortion E as given by Eq. (1) is used directly as the figure of merit (fitness value) of an index assignment and the optimization problem is posed as follows. Find the assignment of the binary indices to the codevectors (or equivalently find the permutation of the codebook) that minimizes Eq. (1).

Formulation of the IA problem in terms of the expected channel distortion according to Eq. (1) is a special case of the quadratic assignment problem (Potter and Chiang, 1995). The QAP is known to be NP-hard (Finke et al., 1987).

3. Genetic annealing search

A general description of genetic algorithms is omitted here but an introduction to the subject can be easily found elsewhere (e.g. Goldberg, 1989; Davis, 1991; Ostrowski, 1995). In the following we present only details relevant to the search method developed.

3.1. Coding of an index assignment

Permutation of a codebook (i.e. a candidate index assignment) is represented directly as a string (chromosome) of the integers (alleles). For instance, the chromosome $|3|5|6|8| \dots$ denotes that the index 3

is assigned to the codevector 0, the index 5 is assigned to the codevector 1, the index 6 is assigned to the codevector 2, etc.

3.2. Genetic annealing

The Genetic annealing algorithm (Price, 1994) is a genetic hybrid algorithm that concerns the evolution of an overlapping population of candidate solutions. This approach possesses an advantage with regard to an implementation on a single processor

machine because of the requirement of a relatively small population (i.e. not more than 50 chromosomes). We propose the following new combination of annealing, selection and recombination mechanisms for the minimization version of GAA.

At the beginning of each *epoch* GAA starts with a population of randomly generated candidate solutions. After that the population is manipulated by using selection, recombination and mutation operators. The parent chromosomes are picked up for the recombination and mutation by using positive selec-

```

1. For  $i := 1$  to number of epochs
  (a) Generate randomly an initial population of  $N$  codebook permutations.
  (b) For  $j := 1$  to  $N$ 
    Compute fitness  $E_j$  of the  $j$ th index assignment.
  End-for
  (c) For  $j := 1$  to  $N$ 
    Initialize the  $j$ th threshold  $Th_j$  with the fitness  $E_j$  of the  $j$ th index assignment:  $Th_j := E_j$ .
  End-for
  (d) For  $j := 1$  to number of generations
    Empty the collected energy bank:  $\Delta E := 0$ .
    For  $k := 1$  to number of recombinations
      Select two chromosomes for the recombination applying twice the positive selection.
      Select a chromosome  $Sub$  for the substitution by an offspring using negative selection.
      If  $\text{random}(0, 1) < \text{recombination rate}$ 
        Recombine solutions selected for the recombination and create one offspring  $Off$ .
        If  $\text{random}(0, 1) < \text{mutation rate}$ 
          Mutate  $Off$ .
        End-if
      Else
        Create  $Off$  by applying mutation to one of the solutions selected for the recombination.
      End-if
      Compute fitness  $E_{off}$  of the resulting offspring.
      If  $E_{off}$  is less than or equal to the threshold  $Th_{sub}$  of the  $Sub$ 
        Add the energy difference to the energy bank:  $\Delta E := \Delta E + Th_{sub} - E_{off}$ .
        Reset the threshold  $Th_{off}$  of the  $Off$ :  $Th_{off} := E_{off}$ .
        Replace  $Sub$  by  $Off$ .
      End-if
    End-for
    If  $\Delta E = 0$  or population is uniform
      Reinitialize population.
    Else
      For  $k := 1$  to  $N$ 
        Anneal the population:  $Th_k := Th_k + C * \Delta E / N$ .
      End-for
    End-if
  End-for
2. Return the best index assignment.

```

Fig. 1. The genetic annealing algorithm.

1. Pick up randomly two chromosomes from the population.
2. If $\text{random}(0, 1) < \text{evolutionary pressure}$
 - Compare the fitness of solutions chosen in the step 1. In the case of different fitness select a better solution; otherwise, select a solution randomly.
- Else
 - Compare the fitness of solutions chosen in the step 1. In the case of different fitness select a worse solution, otherwise select a solution randomly.
- End-if

Fig. 2. The positive selection routine.

tion, which favours better solutions. In turn, for the substitution by an offspring (i.e. a solution that results from the recombination and mutation) the chromosomes are chosen by using negative selection, which favours worse solutions. The duration of a generation is determined by an a priori assumed number of recombinations, which in our case is equal to the population size. To avoid premature convergence of the algorithm, the population is cooled or annealed. This proceeds as follows. Each chromosome, apart from its fitness value (i.e. expected channel distortion) has assigned a *threshold*, which can be larger than the fitness. A chromosome is allowed to be substituted by an offspring only if the fitness value of that offspring is equal to or less than the chromosome's threshold. The difference between the fitness of the offspring and the threshold of the solution substituted by it is called (using a thermodynamic metaphor) an *energy difference*. Each modification of the population results in the addition of the energy difference to the *energy bank* ΔE and resetting of the recombinant's threshold to its fitness. If the collected energy after a pass of the generation equals zero, or the population is uniform (i.e. all fitnesses are equal) the population is reinitialized; otherwise, the process of annealing is performed. A fraction of the collected energy, determined by a cooling constant C , is distributed uniformly over all candidate solutions, increasing their thresholds and the energy bank is emptied. The current epoch ends after the predetermined number of generations. Next, the population is reinitialized and a new epoch starts. This enables to run the algorithm several times, each time with different initial populations, and to choose the best result. In Fig. 1 we present the minimization version of the GAA in a more formal manner. The routine $\text{random}(0, 1)$ returns uniformly distributed random real numbers in the interval $[0, 1)$.

3.3. Positive and negative selection

As the selection mechanism a tournament selection is applied (Goldberg, 1989). In the case of positive selection two chromosomes are picked up randomly from the population. Subsequently for the recombination, with a probability determined by an evolutionary pressure, a better solution is chosen, i.e. a *tournament winner*. Negative selection is performed in a similar manner but in that case a worse fitness is preferred. The typical value of the evolutionary pressure is 0.85. The basic positive selection routine is depicted in Fig. 2.

3.4. Recombination and mutation

For the purpose of recombination a modified version of Partially Matched Crossover (PMX) is used (Buckles et al. 1990). Preparatory to applying the PMX crossover, two chromosomes are drawn from the population by using positive selection as described above. Next, a crossing site is chosen ran-

	Step 1
Parent 1	0 1 2 3* 4 5
Parent 2	5 4 3 2* 1 0
	Step 2
Offspring	3*
	Step 3
Offspring	5 4 3 3* 1 0
	Step 4
Offspring	5 4 2 3* 1 0

Fig. 3. An example of PMX crossover. One crossing site has been chosen that is denoted by the symbol *.

```

For  $i := 1$  to number of crossing sites
1. Choose randomly a crossing site.
2. Copy the allele from the crossing site of the first parent into the same position of an offspring.
3. Copy the alleles of the second parent corresponding to the positions other than crossing site into the corresponding positions of the offspring.
4. Copy the allele of the second parent at crossing site into non-crossing site of the offspring occupied by the same allele as that in the offspring at crossing site.
5. If the number of crossing sites is greater than one
    Substitute second parent by the offspring.
End-if
End-for

```

Fig. 4. The PMX crossover routine.

domly. The allele from the first parent that falls into the selected crossing site is copied into the same position of the offspring. The remaining alleles of the offspring are determined by the second parent as follows. First, alleles of the second parent at the positions other than the crossing site are copied into the corresponding positions of the offspring. Next, the allele of the second parent at the crossing site is placed into the offspring at the non-crossing site occupied by the same allele as that in the offspring at the crossing site. In the example depicted in Fig. 3 the crossing site is denoted by *. The above process can be repeated recursively as shown in Fig. 4.

A mutation of the offspring swaps alleles at two randomly chosen positions.

After the offspring has been created by performing the recombination and mutation operations, the population is checked for diversity, i.e. to have no identical chromosomes because they do not introduce any extra information. The offspring is accepted only if it does not have a copy in the population; otherwise, the selection, recombination and mutation steps are repeated.

The elitist strategy is used, i.e. the best solution obtained during each generation is always preserved and introduced into the evolving population if its copy was destroyed.

4. Computational study

It is very difficult to evaluate objectively the performance of a genetic algorithm. We have decided to approach this problem in a three-fold manner; computation of the index assignment for the

LSF quantization of a speech coder; computation of the index assignment for the vector quantizers trained on the Gaussian data; and finally tests on standard QAP benchmarks.

The implementation of the algorithm has been performed on the DEC alpha workstation. The parameters of the GAA, if not mentioned otherwise, have been adjusted as follows. Population size 50, number of epochs 10, number of generations 10000, recombination rate 0.90, mutation rate 0.15, and cooling constant 0.9850.

4.1. Index assignment for LSF quantization of speech coder

The first experimental example concerns the index assignment problem in the vector quantization of a linear predictive filter in speech coding, see e.g. Knagenhjelm (1993). An all-pole form of the linear predictive filter is employed in most speech coding algorithms of interest today. The filter is usually transformed into the line spectrum frequency representation, and these parameters are quantized and transmitted to the decoder as a part of the bit stream. In this example we examine IA in the context of the TIA/EIA/IS-641 speech coder (TIA/EIA/IS-641, 1996), which has been selected as an enhanced quality speech coder for the digital cellular system IS-136 in the United States. The TIA/EIA/IS-641 is used as a representative example of a modern speech coder and the presented results do not rely on any specific property of this coder.

The TIA/EIA/IS-641 utilizes an all-pole filter of degree 10. The parameters of the filter are computed once in a frame of length 20 ms. The filter is

Table 1

Objective speech quality of the IS-641 speech coder measured by the segmental SNR [dB] as a function of the bit error probability e_c . The channel coding has been disabled. The index assignments were computed assuming MEE and the BEP $e_k = 0.1$ for all bits

e_c	IS-641 assignment	GAA assignment l_1 norm	GAA assignment squared Euclidean norm
0.001	93.65	94.13	91.25
0.005	77.64	78.52	76.59
0.010	64.68	66.34	64.42
0.020	46.72	48.89	47.09
0.040	25.94	28.90	27.53
0.060	15.46	18.40	17.35
0.080	9.82	12.84	11.88
0.100	6.69	9.49	8.58

transformed into the line spectrum frequency representation, and LSF are quantized in three parts by using a split vector quantizer. The quantizer utilizes a moving average predictor. Since the prediction introduces memory into the quantizer, Eq. (1) is not an exact expression of the expected channel distortion. Here Eq. (1) is used as an ad hoc approximation, and theoretical aspects are ignored. For simplicity, we examine merely the first split of the quantizer. This comprises 2^8 three-dimensional codevectors. In TIA/EIA/IS-641 the eight bits of an index are protected against channel errors by the channel coding and Cyclic Redundancy Check (CRC), which validates the correctness of the bits in the channel decoder. Therefore, a good index assignment is not essential in the implementation of this speech coder. However, the example illustrates the potential advantages of IA as a non-redundant channel coding.

The distortion in the synthesized speech is measured by the segmental Signal to Noise Ratio (SNR), see e.g. Quackenbush et al. (1988). The distortion caused by the channel errors is simulated by toggling bits in the transmitted bit stream with a given probability e_c when the channel coding is disabled. Both

the corrupted and the clean bit streams are decoded and the segmental SNR between these two synthesized speech signals is computed. We used the segmental SNR with segments of 128 samples. This corresponds to 16 ms when the sampling frequency is 8 kHz. A speech sample of 25,000 frames, i.e. about 8 min. 20 s, containing American English sentences spoken by male and female speakers is employed as test material.

The index assignment used by the first LSF codebook of TIA/EIA/IS-641 is compared to the two optimized index assignments in Table 1. The index assignments were optimized by using the l_1 norm and the squared Euclidean norm. The MEE assumption and the BEP $e_k = 0.1$ for all bits were used in both cases. The GAA computed IAs result in 2–3 dB gain when e_c increases. In this case the l_1 norm gives a better performance than the squared Euclidean norm.

4.2. Index assignment for vector quantizers trained on the Gaussian data

We have used also a Gaussian VQ-base (Knagenhjelm, 1993) to verify experimentally our

Table 2

Theoretical expected channel distortion for the codebook trained on the Gaussian data. Codebook contains 2^8 three-dimensional codevectors. The index assignments were computed assuming MEE and BEP $e_k = 0.1$ for all bits

Distance measure determined by	Random assignment	Original assignment	GAA assignment
l_1 norm	414	263	250
Squared Euclidean norm	601	236	235

Table 3
GAA assessment by using the QAP benchmarks

Problem	Problem size	Best known solution	Solution found by GAA
Tai256c	256	44951708	44919020
Tai150b	150	499348972	506919687
Esc128	128	64	64
Nug30	30	6124	6124

approach to IA. The database is available on <http://www.it.chalmers.se/> and contains several codebooks constructed for Gaussian sources. The codevectors are arranged so that the indexing is reasonably robust against channel errors. The IAs for the quantizers included in the database have been obtained by using LISA (Knagenhjelm, 1996), i.e. under the assumptions of MEE and that the distance measure between the codevectors is determined by the squared Euclidean norm.

In Table 2 the comparison of the original and GAA computed IAs for the codebook containing 2^8 three-dimensional codevectors (data file k3_08_05.txt) trained on the correlated Gaussian source (correlation 0.5) is presented. The MEE assumption and the BEP $e_k = 0.1$ for all bits were used. We conclude that it is much easier to obtain better than the original assignments with regard to a distance measure defined according to the l_1 norm than the squared Euclidean norm.

4.3. Performance on QAP benchmarks

Finally, the GAA has been tested on standard QAP benchmarks. The QAP database is available on <http://www.diku.dk/~Ekarisch/qaplib/>. The GAA improved the best known solution for the Tai256c problem (Taillard, 1994) that is now 44919020 and found the solutions equal to the best known for the Esc128 (Eschermann and Wunderlich, 1990) and Nug30 (Nugent et al., 1968) problems. For the Nug30 problem the number of epochs and the number of generations were adjusted to 10 and 1000, respectively. The results for some other problems, e.g. Tai150b (Taillard, 1994) were very close to the best known feasible solutions. Table 3 summarizes the outcomes of the QAP experiments.

5. Conclusion

We presented the genetic annealing algorithm that can be applied to a broad area of combinatorial optimization problems, for instance QAP. The results of the experiments confirm that the GAA is useful for the index assignment in vector quantization.

Acknowledgements

This research work has been performed under the Technology Development Centre of Finland (TEKES) research program “Adaptive and Intelligent Systems Applications” – research project “Adaptive and Intelligent Image and Signal Analysis”. T. Ostrowski appreciates leave-of-absence from Technical University of Szczecin.

References

- Buckles, B.P., F.E. Petry and R.L. Kuester (1994). Schema survival rates and heuristic search in genetic algorithms. In: B.P. Buckles and F.E. Petry, eds., *Genetic Algorithms*. IEEE Computer Society Press, Washington, 86–91.
- Davis, L. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.
- Eschermann, B. and H.J. Wunderlich (1990). Optimized synthesis of self-testable finite state machines. In: *Proc. 20th International Symposium on Fault-Tolerant Computing (FTTCS 20)*. Newcastle upon Tyne.
- Farvardin, N. (1990). A study of vector quantization for noisy channels. *IEEE Trans. on Information Theory* 36 (4), 799–809.
- Finke, G., R.E. Burkard and R. Rendl (1987). Quadratic assignment problems. *Annals of Discrete Mathematics* 31, 61–82.
- Goldberg, D.E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, MA.

- Gersho, A. and R.M. Gray (1992). *Vector Quantization and Signal Compression*. Kluwer Academic Publishers, Dordrecht.
- Knagenhjelm, P. (1993). Competitive learning in robust communication. Doctoral Dissertation, Department of Information Theory, Chalmers University of Technology, Goeteborg, Sweden.
- Knagenhjelm, P. (1996). Private communication.
- Knagenhjelm, P., and E. Agrell (1996). The Hadamard transform – A tool for index assignment. *IEEE Trans. Inform. Theory* 42 (4), 1139–1151.
- Koopmans, T.C. and M.J. Beckmann (1957). Assignment problems and the location of economic activities. *Econometrica* 25, 53–76.
- Linde, Y., A. Buzo and R.M. Gray (1980). An algorithm for vector quantizer design. *IEEE Trans. Commun.* 28 (1), 84–85.
- Nugent, C.E., T.E. Vollman and J. Ruml (1968). An experimental comparison of techniques for the assignment of facilities to locations. *Operations Research* 16, 150–173.
- Ostrowski, T. (1995). Computing with genetic algorithms in the context of adaptive neural filtering. *Pattern Recognition Letters* 16 (2), 125–132.
- Potter, L.C. and D. Chiang (1995). Minimax non-redundant channel coding. *IEEE Trans. on Communications* 43 (2–4), 805–811.
- Price, K.V. (1994). Genetic annealing. *Dr. Dobbs' Journal* 10, 128–132.
- Quackenbush, S.R., T.P. Barnwell and M.A. Clements (1988). *Objective Measures of Speech Quality*. Prentice Hall, Englewood Cliffs, NJ.
- Taillard, E.D. (1994). Comparison of iterative searches for the quadratic assignment problem. *Location Science*, To appear.
- TIA/EIA/IS-641 (1996). *TDMA Cellular/PCS – Radio Interface – Enhanced Full-Rate Speech Codec*. Telecommunications Industry Association/Electronic Industry Association/Interim Standard 641, Global Engineering Documents.
- Zeger K., and A. Gersho (1990). Pseudo-Gray coding. *IEEE Trans. Commun.* 38 (12), 2147–2158.