

Optimal Codebook Design Based on Ant Colony Clustering and Genetic Algorithms*

Zhaoan Su, Chundi Xiu

Beihang University, Beijing, China
zhaoan_su@163.com, xcd@buaa.edu.cn

Abstract - Codebook design plays an important role in the performance of signal processing based on vector quantization (VQ) just as speech coding, data compression and pattern recognition. LBG is one of the most effective algorithms and is widely used in codebook generation. Some problems still exist while its performance is remarkable. The LBG algorithm is easy to fall into local optimum. Usually there is strong correlation between the best solution and the initial selection for codebook design. It means that the quantization performance of codebooks from the same training data may varies in a certain range. There is also a certain probability for the algorithm to generate empty voronoi cell. In order to solve these problems, a novel algorithm based on ant colony clustering algorithm and genetic algorithm is proposed in this paper. The new algorithm takes advantage of the excellent global optimal searching ability of genetic algorithm. At the same time, the ant colony clustering algorithm is combined into the process. The dynamic change of the searching direction is adopted during crossover stage. The simulation results of line spectrum frequency parameters in mixed linear excitation prediction (MELP) show that the proposed algorithm is more efficient in its quantization performance compared to that of the LBG and genetic algorithms. Meanwhile, it has good stability in quantization performance.

Index Terms - Codebook design, Vector quantization, Genetic algorithm, Ant colony clustering algorithm.

I. Introduction

Low bit rate speech coding schemes are needed in communication environments with stringent spectrum resources constraints. Vector quantization, popularly known as VQ, plays an important role in low bit rate speech coding area. VQ is an efficient approach, which maps a sequence of signal called vector onto a small set of similar vectors. The set is called the codebook and each individual in the codebook is called a code-word. The design of codebook is the heart to vector quantization's effectiveness.

Many algorithms have been proposed to generate a codebook from the training data. The LBG algorithm, which is proposed by Linde, Buzo and Gray in 1980, has been widely used over the past decades^[2]. It doesn't need to get the probability distribution of the training data and obtains the optimal codebook by constantly classifying the training vectors and computing the new codebook. However, the LBG algorithm has two main drawbacks: (1) the optimal codebook is closely related to the initial selection; (2) it is easy to fall

into local optimum and can produce empty voronoi cell with a certain probability.

In order to solve these problems, many novel algorithms are put forward, such as ant colony clustering algorithm, genetic algorithm (GA) and kernel fuzzy learning algorithm^[2-6]. In Reference [2] ant colony clustering algorithm is used to generate codebook. Because of that the clustering number is automatically formed, nearest neighbor criterion or decomposition method is adopted. Simulation results show that the decoding quality of AWR-WB is not almost degraded adopting new algorithm. A new codebook design method based on genetic and LBG algorithms is presented in Ref [3]. It applies LBG clustering algorithm into genetic algorithm to optimize cluster center. The hybrid method not only improves the quality of codebook but also speed algorithm convergence. Reference [4] puts forward a novel codebook generation algorithm using a combined scheme of principal component analysis and genetic algorithm. The combined scheme makes full use of the near global optimal searching ability of GA and the computation complexity reduction of PCA. Experimental results indicate that the proposed algorithm outperforms the popular LBG algorithm in terms of computational efficiency and quantization performance. However, the existing schemes still cannot generate codebook which has both better quantization performance and good stability.

In this paper a novel codebook design algorithm based on ant colony clustering algorithm and genetic algorithm (ACGA) is proposed. In this algorithm, firstly, ants of different number are used to cluster the training data in two-dimensional plane of different size. The centroid of the cluster is calculated as the initial vector of the codebook. Secondly, the codebooks are considered as the initial individuals in genetic process. In order to speed up the convergence, dynamic searching direction is used in crossover stage. Until the best solution gets the quantization distortion that is smaller than the threshold, the iteration work will be stopped.

This paper proceeds as follows. Section 2 briefly reviews the ant colony algorithm. Section 3 gives a summary of genetic algorithm and introduces its drawbacks. Details of new algorithm ACGA is presented in section 4. Section 5 contains the experimental results and analysis of the applications of ACGA in low bit rate speech coding. Lastly, this paper is summarized in Section 6.

* This work was supported by the National Natural Science Foundation of China (Grant No. 61272501), the National Key Basic Research Program (NKBRP) (973 Program) (Grant No. 2012CB315905) and the Beijing Natural Science Foundation.

II. Ant Colony Clustering Algorithm

As one kind of swarm intelligence algorithm, ant colony clustering algorithm, proposed by Italian scholar A. Dorigo, is inspired by the behaviors of the ants. It doesn't depend on a specific mathematical description of the problem and has the ability of global optimization. In nature, the ant heap is attractive to the worker ant which is carrying a dead ant. Usually, the larger the size of the ant heap is, the more attractive the ant heap is to the worker ant; the smaller the size of the ant heap is, the less attractive the ant heap is to the worker ant. This program referred to LF algorithm is proposed by Lumer and Faieta. B.

The main idea of LF algorithm is to randomly distribute the training data to two-dimensional plane within a limited range. Then a certain number of ants are selected to divide the data into different clusters. For each ant, the probability to pick-up or drop data is determined by probability functions. Both are functions of local density. Then a random number is generated. Depending on the random number, the state of ant is determined and the data are clustered. Here are the parameters used in the clustering process.

O_i : The i^{th} training set to be clustered

$Z * Z$: The limited two-dimensional plane

$f(O_i)$: The local density of O_i

$P_p(O_i)$: The picking probability

$P_d(O_i)$: The dropping probability

The local density $f(O_i)$ is defined as the similarity between the data in limited neighbor area. It's calculated as (1):

$$f(O_i) = \max \left\{ 0, \frac{1}{s^2} \sum_{O_j \in \text{Neigh}(s*s)} \left[1 - \frac{d(O_i, O_j)}{\alpha} \right] \right\} \quad (1)$$

Where $d(O_i, O_j)$ represents the Euclidean distance between O_i and O_j in initial space, $s*s$ represents the space for data searching and s is the search radius. α is the dissimilarity factor.

$P_p(O_i)$ and $P_d(O_i)$ are calculated as (2) and (3) respectively.

$$P_p(O_i) = \left(\frac{k_1}{k_1 + f(O_i)} \right)^2 \quad (2)$$

$$P_d(O_i) = \begin{cases} 2f(O_i), & \text{when } f(O_i) < k_2 \\ 1, & f(O_i) \geq k_2 \end{cases} \quad (3)$$

Where k_1 and k_2 are threshold constants.

Because that the ant randomly makes a decision to pick up or drop the data, the convergence speed of the LF algorithm is slow. Sometimes, under the restrictive conditions of limited ants and clustering time the LF algorithm can't reach a converged state. But it has the advantage that initial selection

is not needed in the process. This can effectively weaken the correlation between the initial selection and the final solution.

III. Genetic Algorithm

Genetic algorithm (GA) is a computing model which simulates the natural selection and gene mutation in Darwin's theory of biological evolution. This is originally proposed by Professor J. Holland from the Michigan University in 1975. GA can effectively avoid falling into local optimum with its excellent implicit parallelism and better global optimization. In the process, the algorithm can adaptively adjust the searching direction and mutate at a certain probability.

The genetic algorithm procedure includes four stages: calculation of the individual's fitness, natural selection, individual crossover and gene mutation. In the process of codebook design with GA, the individual is the codebook which contains a certain number of vectors. The vector is called the chromosome in individual and the number of the chromosomes is equal to the codebook size. Gene on the chromosome represents the signal on each dimension and the chromosome length is the same to the signal dimension. The population consists of N individuals. N is called the size of the population.

The codebook training process is shown as follows.

a) Initialization: firstly, set codebook size M , gene mutation probability p_m , crossover probability p_c , population size N , chromosome length L and number of training data. Secondly, M vectors are selected randomly from the training data as the initial individuals.

b) Calculation: calculate the fitness f_i of each individual in the population as (4).

$$f_i = \frac{1}{d_i^2} \quad (4)$$

Where d_i represents the distortion while the training data are quantized with the codebook respectively. d_i is calculated as (5).

$$d_i = \frac{1}{N} \sum_{j=1}^N \sum_{l=1}^L (x_{i,j} - \hat{x}_{i,j})^2 \quad (5)$$

Where $x_{i,j}$ and $\hat{x}_{i,j}$ represent the original data in individual and the quantized vector in codebook respectively.

c) Selection and Crossover: Depending on fitness all individuals are rearranged in descending order. Individual having large fitness is considered to be more fit than individual having small fitness. Thus, p_c percent of the population are considered as fit individuals and carried to the next generation. After that the remaining individuals are generated by crossover operation. The first individual and the last individual are paired using single point crossover method. The procedure continues until enough individuals are generated.

d) Mutation: In GA, mutation process is taken to avoid falling into local optimum during the generation of new individuals. According to the mutation probability p_m , $N * p_m$ individuals are selected.

e) Termination: Usually two termination criteria are used in the process. Either the process is executed to produce a fixed number of generations and the individual that has the largest fitness is considered as the best solution, or there is no more improvement for the best solution.

Although the genetic algorithm can avoid falling into local optimum, its convergence speed is slower than LBG algorithm.

IV. Ant colony-genetic algorithm

Based on ant colony clustering algorithm and genetic algorithm, a novel algorithm for codebook design is proposed in this paper. It's called ant colony-genetic algorithm (ACGA) which effectively avoids falling into local optimum and weakens the correlation between the best solution and initialization. The main process of ACGA is as follows.

Step 1: the initial individuals of the population are generated with LF algorithm. For different individual, training sequences of a fixed number are selected randomly for clustering process. Then the number of ants and the size of two-dimensional plane keep change until enough codebooks are generated.

Step 2: Considering that the cluster number ant colony clustering algorithm C_{out} may be different from the codebook size, the modify operation is conducted. When c is larger than M , the cluster of smallest size is eliminated. When c is smaller than M , firstly find the cluster which includes the most training samples. Secondly, the samples are portioned into two clusters according to nearest neighbor criterion. The work is carried until c fits the codebook size.

Step 3: the original population that consists of the codebooks generated in step two is considered as the first generation in genetic algorithm. Then the process follows the genetic algorithm in section 3. Taken the specific characters of the actual data compressing into consideration, new crossover rule as (6) is adopted.

$$\begin{aligned} X &= a * A + (1 - a) * \tilde{A} \\ Y &= (1 - a) * A + a * \tilde{A} \end{aligned} \quad (6)$$

Where A is the selected individual and \tilde{A} is its partner. a represents the weighted factor which is function of the individual's fitness as (7). The larger the fitness is, the more influence the individual has on the next generation. This seems like to the artificial selection in biological evolution.

$$a = \frac{f_A^2}{(f_A + f_{\tilde{A}})^2} \quad (7)$$

V. Simulation results and analysis

A. Experiment setup

The database of the line spectrum frequency parameters (lsf) in mixed excitation linear prediction (MELP) is used in simulations. The speech signal is sampled at 8 KHz and each frame consists of 180 points. The training set contains 1600 training samples. The simulation works in Windows7 system while using software Matlab of version 2010a. Parameters used in the process are initialized as follows:

$$k_1 : 0.45 \quad k_2 : 0.5$$

The dissimilarity factor $\alpha : 2$

The probability of crossover $p_c : 0.6$

Size of population $N : 20$

The probability of mutation $p_m : 0.35$

Ant number: [25, 28, 31.....82, 85]

Size of two-dimension plane $Z * Z : [100, 102, 104.....136, 138, 140]$

B. Simulation results and analysis

The simulation work is conducted in two steps. The first step is to test the quantization performance of the proposed algorithm. The distortion is calculated as (8).

$$dis = \frac{1}{N} \sum_{j=1}^N \frac{\sum_{i=1}^L (x_{i,j} - \hat{x}_{i,j})^2}{x_{i,j}^2} \quad (8)$$

Experimental data are shown in TABLE 1. It shows that the ACGA is more effective than LBG and GA algorithms in quantization performance.

TABLE 1 the performance of algorithms for lsf.

Codebook size	Quantization dis for training data		
	LBG	GA	ACGA
32	0.098	0.095	0.084
64	0.064	0.058	0.046
128	0.035	0.030	0.019

The second step is to test whether the proposed algorithm gets avoid of the disadvantages of the LBG algorithm. The codebook size is set to 32. Under the same condition iterative works are carried to get codebooks. The calculation of dis is conducted and shown in Fig. 1.

As can be seen from Fig. 1, the quantization performance dis of the same size codebooks which are taken from the training data with ACGA remains substantially unchanged. Compared with the performance of LBG algorithm, conclusion is given out that the ACGA does effectively get avoid of the strong correlations between the best solution and the initial selection.

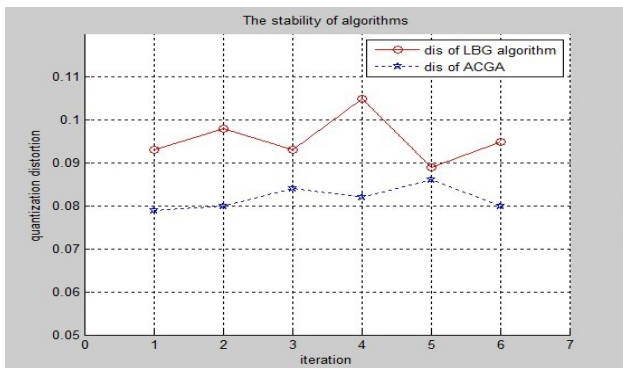


Fig. 1: the stability of different algorithms.

VI. Conclusions

To solve the VQ codebook design problem, a novel algorithm based on ant colony clustering algorithm and genetic algorithm is proposed in this paper. Ant colony clustering algorithm is used to generate the initial individuals for the process of genetic algorithm. After that dynamic change of the searching direction is taken in crossover stage. Experimental results indicate that the new algorithm for the codebook design of the lsf parameters in MELP has better performance in quantization error and stability compared to that of the LBG and GA algorithms.

References

- [1] R. Krishnamoorthy, R. Punidha. Low Bit-rate Multi Stage Vector Quantization based on Energy Clustered Training Set. *Multimed Tools Appl.* 2012, pp: 1244-1260.
- [2] Fenglian Li, Xueying Zhang. The Codebook Design based on Ant Colony Clustering Algorithm and its Application in AWR-WB. *IEEE Conference on Computing, Measurement, Control and Sensor Network.* 2012, pp: 109-112.
- [3] Qian Chen. Vcvector Quantization Method for Image Compression based on GA and LBG Clustering Algorithms. *Computer Science.* 2012, Vol.39: 280-286.
- [4] Hongwei Sun, Kwok, Yan Lam, Siu, Leung Chung. Efficient Vector Quantization using Genetic Algorithm. *Neural Comput & Applic.* 2005, Vol.14:203-211.
- [5] Zongbo Xie, Jiuchao Feng. Codebook Design for Vector Quantization based on a Kernel Fuzzy Learning Algorithm. *Circuits Syst Signal Process.* 2011, Vol.30: 999-1010.
- [6] Dr. H. B. Kekre, Chetan Agarwal. Codebook optimization using Genetic Algorithm and Simulated Annealing. *S.J. Pise(ed), ThinkQuest.* 2011, pp:114-119.
- [7] Stephen So, Kuldip K. Paliwal. Efficient Product Code vector quantization using the switched split vector quantiser. *Journal Digital Signal Processing.* 2007, Vol.17: 138-171.
- [8] Hsiang-Cheh Huang, Jeng-Shyang Pan, Zhe-Ming Lu. Vector quantization based on genetic simulated annealing. *IEEE Journal Signal Processing.* 2001, Vol.81: 1513-1523.
- [9] Shanxue Chen, Jiaguo Wang, Fangwei Li. A New Codebook Design Algorithm of Vector Quantization Based on Hadamard Transform. *Electrical Engineering and Control.* 2011, pp: 181-187.
- [10] Bing Hui, KyungHi Chang. Low Complexity Vector Quantization Algorithm for Codebook Optimization. *IEEE Wireless Communications and Networking Conference.* 2012, pp:868-872.