# Improving Image Vector Quantization with a Genetic Accelerated K-Means Algorithm

Carlos R.B. Azevedo[1,⋆], Tiago A.E. Ferreira[2], Waslon T.A. Lopes[3], and Francisco Madeiro[1]

[1] Catholic University of Pernambuco, Center for Science and Technology, Brazil
{azevedo,madeiro}@dei.unicap.br
[2] Federal Rural University of Pernambuco, Department of Statistics and Informatics, Brazil
tiago@deinfo.ufrpe.br
[3] AREA 1 – College of Science and Technology, School of Electrical Engineering, Brazil
waslon@ieee.org

**Abstract.** In this paper, vector quantizer optimization is accomplished by a hybrid evolutionary method, which consists of a modified genetic algorithm (GA) with a local optimization module given by an accelerated version of the $K$-means algorithm. Simulation results regarding image compression based on VQ show that the codebooks optimized by the proposed method lead to reconstructed images with higher peak signal-to-noise ratio (PSNR) values and that the proposed method requires fewer GA generations (up to 40%) to achieve the best PSNR results produced by the conventional GA + standard $K$-means approach. The effect of increasing the number of iterations performed by the local optimization module within the proposed method is discussed.

## 1 Introduction

Vector quantization (VQ) is a lossy compression technique which plays an important role in many image coding systems, leading to high compression rates [1]. VQ operates according to a minimum distortion rule and can be defined as a mapping $Q$ from an input vector $\boldsymbol{x} \in \mathbb{R}^k$ into a finite subset $W \subset \mathbb{R}^k$ containing $N$ distinct reproduction vectors. Thus, $Q : \mathbb{R}^k \rightarrow W$. Codebook $W = \{\boldsymbol{w}_i\}_{i=1}^{N}$ is a set of codevectors (reconstruction vectors), $k$ is the dimension of the codevectors and $N$ is the codebook size. The mapping $Q$ leads to a partitioning of $\mathbb{R}^k$ in $N$ disjoint regions (also known as Voronoi cells) $S_i$, $i = 1, 2, \ldots, N$, in which each region $S_i$ is defined as $S_i = \{\boldsymbol{x} : \ Q(\boldsymbol{x}) = \boldsymbol{w}_i\}$ or alternatively, $S_i = \{\boldsymbol{x} : \ d(\boldsymbol{x}, \boldsymbol{w}_i) \leq d(\boldsymbol{x}, \boldsymbol{w}_j), \forall j \neq i\}$, where d(·,·) is a distortion measure. Codevector $\boldsymbol{w}_i$ is the representative vector of all input vectors belonging to the corresponding cell $S_i$.

⋆ Author thanks the National Council of Technological and Scientific Development (CNPq) for the financial support.

J. Blanc-Talon et al. (Eds.): ACIVS 2008, LNCS 5259, pp. 67–76, 2008.

In the context of image VQ optimization, a genetic algorithm (GA) can be used to improve the quality of the reconstructed images by performing an evolutionary search for the optimum fixed length set of blocks of pixels which represents the images to be quantized. For this task, codebooks previously obtained with a VQ codebook design algorithm are injected into the initial population of a GA. Seeding a GA's population with known solutions generally provide useful information on the structure of the search space [2]. Hence, the initial population of a GA for image VQ optimization contains reasonable starting codebooks which are to be optimized by means of successive applications of genetic operators in an evolutionary process.

The $K$-means algorithm, also known as Generalized Lloyd Algorithm (GLA) or Linde-Buzo-Gray (LBG) algorithm [5], is the most used technique for codebook design. This paper describes a hybrid genetic accelerated $K$-means algorithm which is applied to optimize codebooks for image VQ previously designed by two versions of $K$-means (standard and accelerated). The proposed GA updates the codebooks produced at the end of each generation according to the method proposed by Lee *et al.* [6], which corresponds to an accelerated version of $K$-means. Also, a partial distance search is integrated on the search for the nearest neighbors performed by the Lee *et al.* method within the proposed GA. The performance of the proposed method is compared to that one of the conventional methodology, which consists on using GA with standard $K$-means. Simulation results concerning image VQ show that the proposed method outperforms the conventional approach in the sense that it leads to better codebooks, which restults in reconstructed images with higher peak signal-to-noise ratio (PSNR), in earlier stages of the evolutionary process.

## 2 The *K*-Means Algorithm

Let the iteration of $K$-means be denoted by $n$. Given $k$, $N$ and a distortion threshold $\epsilon > 0$, the $K$-means algorithm [5] consists of the following steps:

*Step 1* (initialization). Given an initial codebook $W_0$ and a training set $X = \{\boldsymbol{x}_m; m = 1, 2, \ldots, M\}$, set $n = 0$ and $D_{-1} = \infty$.
*Step 2* (partitioning). Let $W_n$ be the codebook at the $n$-th iteration and $\boldsymbol{w}_i^n$ the $i$-th codevector on $W_n$. Assign each training vector (input vector) in the corresponding class (Voronoi cell) according to the nearest neighbor rule; determine the distortion

$$D_n = \sum_{i=1}^{N} \sum_{\boldsymbol{x}_m \in S_i} d(\boldsymbol{x}_m, \boldsymbol{w}_i^n). \tag{1}$$

*Step 3* (convergence test). If $(D_{n-1} - D_n)/D_{n-1} \leq \epsilon$ then stop, with $W_n$ representing the final codebook (designed codebook); else, continue.
*Step 4* (codebook updating). Calculate the new codevectors as

$$\boldsymbol{w}_i^{n+1} = \mathcal{C}(\mathcal{V}(\boldsymbol{w}_i^n)), \tag{2}$$

where $\mathcal{C}(\mathcal{V}(\boldsymbol{w}_i^n))$ is the centroid of the partition $\mathcal{V}(\boldsymbol{w}_i^n)$; set $W_{n+1} \leftarrow W_n$; set $n \leftarrow n+1$ and go to Step 2.

## 3 The Accelerated *K*-Means Algorithm

The algorithm proposed by Lee *et al.* [6] corresponds a modification introduced in Step 4 of $K$-means. In this approach, the new codevector will be updated according to

$$\boldsymbol{w}_i^{n+1} = \boldsymbol{w}_i^n + s(\mathcal{C}(\mathcal{V}(\boldsymbol{w}_i^n)) - \boldsymbol{w}_i^n). \tag{3}$$

This method may be seen as a look ahead approach aiming at improving convergence, while reaching a smaller value of average distortion. In the experiments reported in [6], when the value of the scale $s$ is about 1.8, the algorithm generally achieves good performance. It should be noted that a scale value of $s = 1.0$ implements the standard $K$-means. It is worth to mention that equation 3 resembles the update formula for a neuron on Kohonen's competitive learning scheme [3].

## 4 Partial Distance Search (PDS)

The partial distance search (PDS), proposed by Bei and Gray in [4], is a method for reducing the computational complexity of the nearest neighbor search. The PDS method decides that a codevector is not the nearest neighbor of an input vector if, for some $j < k$ ($k$ is the dimension of the vector quantizer), the accumulated distance for the first $j$ samples of the input vector is greater than the smallest distance previously computed in the search. Then, the computation is terminated for that codevector and the distance computation for the next codevector begins. With this approach, the number of multiplications, additions and subtractions is dramatically reduced in comparison with the full search. Although PDS increases the number of comparisons, the global complexity of the nearest neighbor search is reduced when compared with the full search.

## 5 Hybrid Evolutionary Clustering

The $K$-means algorithm can be regarded as a hill climbing clustering strategy [8]. Hence, it is supposed to achieve local optima, *i.e.* suboptimal partitions. Stochastic approaches such as GA's can be used to find a globally optimal partition, as the search can escape from local optima.

Considering the population of a GA being made up of various codebooks, Fränti [9] presents three possibilities for integrating $K$-means as a local search module on a GA: (i) to apply $K$-means for the output of GA (GA + $K$-means); (ii) to apply GA for the output of $K$-means ($K$-means + GA); (iii) to apply $K$-means for every codebook in the GA population.

It is also possible to specify a GA for VQ without using any integrated hill climbing technique such as $K$-means, but it has been shown that such GA performs worse than if a clustering algorithm is used jointly [9]. Since $K$-means algorithm seeks to satisfy the constraints required by VQ optimality conditions [1], its integration with GA is beneficial [9, 8].

## 6 The Proposed Method

The method proposed in the present paper follows a GA + $K$-means strategy, where an individual is considered to be a codebook composed with $N$ genes, which corrersponds to $N$ codevectors. However, some differences may be pointed out when comparing it with those methods described by Fränti [9]: (i) the GA replacement strategy has been modified to accept all offspring which represents better solutions than those already known; (ii) with probability $1 - p_{acc}$ ($p_{acc}$ is the acceptance probability), the generated solutions which do not obtain gains over the worse individual present in the population are rejected. When accepted, an offspring will always replace the worse individual; (iii) the standard $K$-means algorithm was replaced by the accelerated version proposed by Lee *et al* [6]; (iv) instead of applying the accelerated $K$-means to the best solution in the GA's population at each generation, it is applied to the new offspring.

All modifications introducted by the proposed evolutionary optimizer are explained in the following: the first one depends on the $p_{acc}$ parameter – for $p_{acc} = 0$, this strategy is equivalent to a greedy genetic search [11, 14]; for $p_{acc} = 1$, the strategy becomes the one implemented on the canonical GA [7] (accept all offspring). This parameter gives a fine control of diversity levels in the GA, while guarantees that good solutions will not be rejected. This improvement was benchmarked in Leung *et al.* [12] and it was used for training multilayer perceptron neural networks, as reported in that paper.

The second improvement constitutes the main contribution of this paper. It is expected that this modification will lead to better results in terms of peak signal-to-noise ratio, as shown in the results' section.

Finally, the third modification is justified by the sake of maintaining diversity and as an attempt to achieve better results in earlier stages of the evolutionary process (when compared with the conventional approach): in any GA scheme, the common way of controlling diversity is by changing the mutation rate, *i.e.* the frequency that mutation will be applied to an offspring. In this sense, the accelerated $K$-means integration with GA can be regarded as an additional local optimization module, which acts on the newly candidate solution. Hence, the accelerated $K$-means module improves the newly offspring. The offspring is expected to have a higher probability of being better than its parents. Note that such strategy was successfully utilized in [8] for VQ codebook design. Other mechanisms of the proposed GA are now explained:

*Initialization* – The initial population is the set $\{W_z\}_{z=1}^{psize}$ of initial codebooks trained with $K$-means algorithm with $\epsilon = 10^{-3}$, in which the size of the population is denoted by *psize* and $W_z$ is the $z$-th input codebook.

*Selection for reproduction* – A fitness proportional selection strategy commonly referred to as *roulette wheel* is adopted: the population is ordered by the individuals' fitness and two parents are sampled with probability proportional to their fitness values, within a stochastic simulation scheme.

*Genetic operators* – For parents' recombination, a generalized crossover operator [10] is implemented in the following way: first, a random integer *segsize* is sampled from the set $\{1, \ldots, N/2\}$, in which *segsize* denotes the segment size. Then, for each block composed of *segsize* genes, one chooses, with equal probability, the corresponding block from one of the parents for composing the genes of the offspring. It can be seen that this algorithm is equivalent to uniform crossover when $segsize = 1$ and to one-point crossover when $segsize = N/2$ with the *locus* point on the half of the chromosome. For mutation, let $\mu$ be a random variable uniformly distributed on the range $[0.8, 1.2]$. Then, the scalar product of $\mu$ with a random chosen gene is performed with probability *pmut*.

## 7 Results

The coding performance of all optimization methods was evaluated on $256 \times 256$ pixels, monochrome images, originally encoded at 8 bpp: Boat, Barbara, Clock, Elaine, Goldhill, Lena, Mandrill, Peppers and Tiffany (Figure 1). In all simulations, for a given image and a fixed $N$, the same initial codebooks were used. The distortion threshold $\epsilon = 10^{-3}$ was assumed for the designing of these codebooks with both $K$-means and Lee *et al.* algorithms.

For each combination of image and codebook size, two distinct populations composed by 20 codebooks (*psize*=20) were generated: the first, composed by codebooks designed with the standard $K$-means and the second by codebooks designed with Lee *et al.* algorithm with a fixed scale of $s = 1.8$. For the proposed method (GA + accelerated $K$-means), $p_{acc}$ was set to 0.1, while the number of iterations performed by the local optimization module ranged from $n = 1$ to $n = 3$. The common parameters used on both proposed and conventional (GA + standard $K$-means) methods were: mutation rate of 0.2 and a maximum of 500 generations as the stopping criterion. The fitness function is assumed to be the peak signal-to-noise ratio (PSNR) [1], which is also used to assess the objective quality of the reconstructed images.

Vector quantization with dimension $k$=16 was considered, corresponding to the use of blocks of $4 \times 4$ pixels. Codebook sizes of $N$=32, 64, 128, 256 and 512 were considered, corresponding to coding rates of 0.3125, 0.375, 0.4375, 0.5 and 0.5625 bpp.

**Fig. 1.** Image data set used in the experiments

**Table 1.** Average PSNR (dB) for Lena obtained in 50 experiments

| $N$ | $K$-means | GA + $K$-means | Proposed Method |
|---|---|---|---|
| 32 | 26.59 | 26.69 | 26.70 |
| 64 | 27.71 | 27.91 | 27.91 |
| 128 | 28.80 | 29.07 | 29.11 |
| 256 | 29.90 | 30.55 | 30.73 |
| 512 | 31.11 | 32.33 | 32.62 |

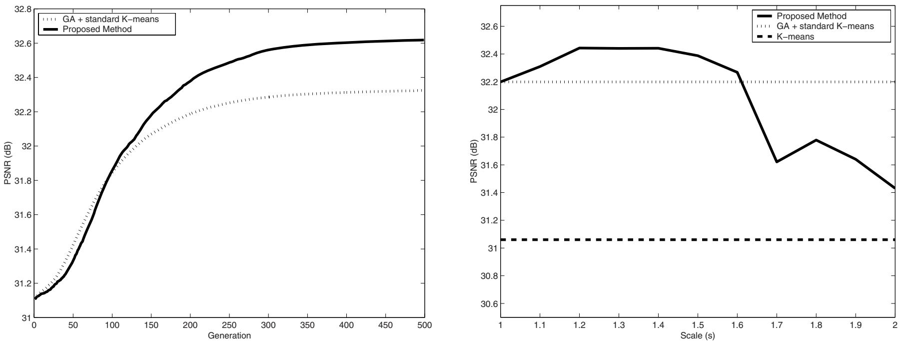
Table 1 summarizes the experimental results obtained after 50 runnings of the GA + standard $K$-means approach and the proposed method (with $s = 1.5$) for Lena. For the results reported in the table, the initial population was designed with the standard $K$-means.

From Table 1, it can be noted that the proposed method outperforms the conventional techniques investigated. In addition, as the codebook size ($N$) raises, the gain achieved with the proposed GA increases. It should be noted that the $K$-means column on Table 1 refers to the PSNR values derived from the population ($K$-means designed codebooks) which fed both GA's versions.

An experimental study was also conducted for adjusting the scale value ($s$) used on the proposed method. For this task, scale values from the set $\{1.1, 1.2, \ldots, 2.0\}$ were tested for each combination of image and codebook size ($N$). The results for the best values and the respective improvements given in terms of PSNR measure in comparison with the initial codebooks (initial population) for 0.5625 bpp encoded images are reported on Table 2. The best results for the proposed method were achieved for Clock, Lena, Peppers, Tiffany, Boat and Elaine images: in comparison with the best codebook present on the initial population, the average gains for those images ranged from 1.10 to 1.58 dB, while in comparison with the average PSNR values obtained with the GA + standard $K$-means approach, the average gains ranged from 0.20 to 0.33 dB.

**Table 2.** Average improvements obtained by the proposed method over both $K$-means and GA + $K$-means for the best scale values ($s$) on images quantized at 0.5625 bpp

| Image | Gain (dB) | | Gain (%) | |
|---|---|---|---|---|
| | $K$-means | GA + $K$-means | $K$-means | GA + $K$-means |
| Barbara ($s = 1.2$) | 0.62 | 0.15 | 1.74 | 0.50 |
| Boat ($s = 1.4$) | 1.12 | 0.21 | 3.85 | 0.72 |
| Clock ($s = 1.1$) | 1.58 | 0.33 | 5.12 | 1.03 |
| Elaine ($s = 1.3$) | 1.10 | 0.20 | 3.36 | 0.58 |
| Goldhill ($s = 1.2$) | 0.50 | 0.00 | 1.62 | 0.01 |
| Lena ($s = 1.5$) | 1.50 | 0.29 | 4.86 | 0.90 |
| Mandrill ($s = 1.1$) | 0.39 | 0.05 | 1.49 | 0.19 |
| Peppers ($s = 1.2$) | 1.49 | 0.26 | 4.82 | 0.81 |
| Tiffany ($s = 1.5$) | 1.47 | 0.27 | 4.39 | 0.79 |




(a) Comparison of the two methods (b) Scale variation study

**Fig. 2.** (a) Average evolution of the PSNR for Lena with $N = 512$ after 50 trials. (b) Effect of scale variation in the PSNR for Peppers with $N = 512$.

Figure 2(a) shows an example of the evolution performed by the proposed method and by the GA + standard $K$-means approach for Lena encoded at 0.5625 bpp. It is observed that, at the end of 500 generations, the proposed method reaches an average PSNR value of 32.62 dB (after 50 runnings) for the reconstructed image, while the GA + standard $K$-means method achieved a 32.33 dB average PSNR value. The figure also shows that, after about 257 generations, the average PSNR value obtained by the GA + standard $K$-means method stabilizes. The proposed method, by its turn, requires only 161 generations to reach the same result (32.33 dB) obtained by the conventional method. It is observed that the proposed method has achieved an average reduction of 37% regarding the number of necessary generations to reach that PSNR result.

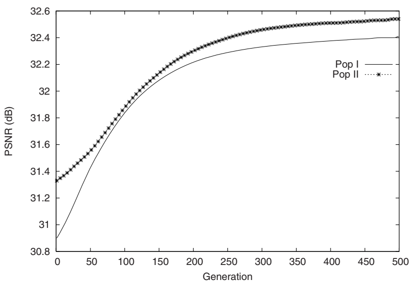
Figure 2(b) shows the effect of varying the scale parameter for Peppers encoded at 0.5625 bpp. The results show that the proposed method obtains, for $s \leq 1.6$, PSNR average values better than the ones obtained with the GA + standard $K$-means method.

Table 3 shows the average PSNR values for Clock and Tiffany images reconstructed from 40 codebooks present in two distinct initial populations: the first one designed with the $K$-means (*Pop I*) and the second one with Lee *et al.* algorithm (*Pop II*). It can be noted that the codebooks belonging to population *Pop II* lead to higher average PSNR values for the reconstructed images, when compared to those of *Pop I*.

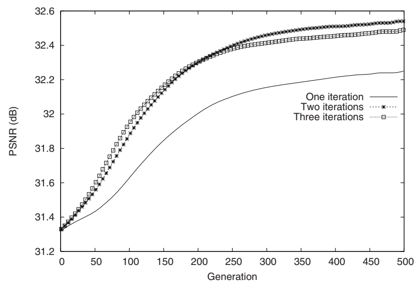
Figure 3(a) shows the effect of initializing the proposed method with both populations (*Pop I* and *Pop II*) for Clock encoded at 0.5625 bpp, considering two iterations of Lee *et al.* performed by the local optimization module at each GA generation. From Figure 3(a), it is observed that the curve which represents the average PSNR of the population *Pop II* starts above the curve of the population *Pop I*. At the course of the evolutionary process, on average, it is observed that the evolution of the population *Pop I* produces, at the 125th generation, individuals with average fitness equivalent to those produced by population *Pop II*.

**Table 3.** Average PSNR (dB) of the initial populations for Clock and Tiffany images encoded at 0.5625 bpp

| Imagem | $N$ | Population | |
|---|---|---|---|
| | | *Pop I* | *Pop II* |
| Clock | 256 | 29.47 | 29.77 |
| | 512 | 30.90 | 31.32 |
| Tiffany | 256 | 32.40 | 32.72 |
| | 512 | 33.61 | 34.15 |




(a) Varying the initial population




(b) Varying the number of iterations

**Fig. 3.** (a) Average evolution for the proposed method using both initial populations for Clock encoded at 0.5625 bpp. (b) Average evolution of the proposed method varying the number of iterations of the Lee et al. for Clock encoded at 0.5625 bpp.

**Table 4.** Average PSNR (dB) obtained for Clock with the proposed method by varying the number of Lee *et al.* iterations ($n$), the codebooks' size ($N$) and the initial population

| Population | $N = 256$ | | | $N = 512$ | | |
|---|---|---|---|---|---|---|
| | $n = 1$ | $n = 2$ | $n = 3$ | $n = 1$ | $n = 2$ | $n = 3$ |
| *Pop I* | 30.45 | 30.65 | 30.71 | 32.47 | 32.41 | 32.45 |
| *Pop II* | 32.46 | 32.75 | 32.77 | 32.25 | 32.54 | 32.49 |

However, it can be noted that, as the number of generations increases, the gain obtained with the population *Pop II* grows and ends with a 0.13 dB gain, after 500 generations.

It is observerd in Table 4 that the PSNR values seem to increase with the number of iterations ($n$). However, for most cases, the PSNR results obtained with $n = 3$ iterations are slightly better than the ones obtained with two iterations.

A study of the effect of increasing the number of Lee *et al.* iterations within the proposed method is shown in Figure 3(b) for Clock encoded at 0.5625 bpp. Based on those results, two applications of the local optimization module are recommended within the proposed method with the purpose of obtaining better codebooks in earlier stages of the evolutionary search. Inasmuch as the GA is not guaranteed to obtain significant gains in terms of PSNR when more than two applications of the local optimization module is performed, the usage of more than two iterations of the Lee *et al.* method within the proposed method is not recommended. Besides, the tiny gains for PSNR values obtained with three or more applications of the local module does not justify the computational cost of the additional iterations.

Finally, it is observed that the integration of the PDS algorithm with the proposed method – specifically at Step 2 of the accelerated $K$-means algorithm within the local optimization module – has reduced the mean running time approximately by 15%. As an example, for Tiffany enconded at 0.5 bpp, the method has spent 233 seconds, in average, when the search for the nearest neighbors was performed with the full search (FS), while it has spent 204.2 seconds with the usage of the PDS algorithm, considering all 500 generations.

## 8 Conclusion

This paper presented a hybrid evolutionary approach for image VQ codebook optimization through the usage of Lee *et al.* iterations instead of standard $K$-means as a new local optimization module on a modified GA. Results have shown that the method is promising in terms of improving the initial codebooks obtained by standard VQ codebook design algorithms. More specifically, when compared with conventional GA + standard $K$-means approach, the proposed method produces, in earlier stages of the evolutionary search, better codebooks, which lead to reconstructed images with higher average PSNR values. Moreover, it was observed that two Lee *et al.* iterations lead to the best cost-benefit relation considering running time and performance in terms of PSNR of the reconstructed images. Finally, the benefits of integrating the partial distance search (PDS) algorithm with the proposed method has been asserted as an approach for reducing the computational complexity of the optimizer module as a whole, providing a higher eficiency on the running time required to optimize the input codebooks.

Future studies should consider the investigation of an adequate stop criterion, based on the convergence rate of the algorithm. Also, auto-adaptive methods, which could automatically adjust the scale parameter ($s$), are of great interest. Current works include the investigation of a Terrain-Based Genetic Algorithm (TBGA) [13] for deriving heuristics for the adjustment of the scale parameter during the evolutionary search.

## References

1. Gersho, A., Gray, R.M.: Vector Quantization and Signal Compression. Kluwer Academic Publishers, Boston (1992)
2. Louis, J.S., Johnson, J.: Solving Similar Problems Using Genetic Algorithms and Case-Based Memory. In: Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA 1997), pp. 283–290. Morgan Kaufmann, San Francisco (1997)

3. Kohonen, T.: Self-Organizing Maps. Springer, Berlin (2001)
4. Bei, C.D., Gray, R.M.: An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization. IEEE Trans. on Communications 33, 1132–1133 (1985)
5. Linde, Y., Buzo, A., Gray, R.M.: An Algorithm for Vector Quantizer Design. IEEE Trans. on Communications 28, 84–95 (1980)
6. Lee, D., Baek, S., Sung, K.: Modified K-means Algorithm for Vector Quantizer Design. IEEE Signal Processing Lett. 4, 2–4 (1997)
7. Holland, J.H.: Adaptation in Natural and Artificial Systems: An Introductory Analysis with Application to Biology, Control and Artificial Intelligence. MIT Press, Cambridge (1992)
8. Krishna, K., Murty, M.: Genetic K-means Algorithm. IEEE Trans. on Systems, Man and Cybernetics 29, 433–439 (1999)
9. Fränti, P.: Genetic Algorithm with Deterministic Crossover for Vector Quantization. Pattern Recog. Lett. 21, 61–68 (2000)
10. Ng, W., Choi, S., Ravishankar, C.: An Evolutionary Approach for Vector Quantizer Design. In: Proceedings of the Second IEEE International Conference on Evolutionary Computing, pp. 406–411 (1995)
11. Smith, J.: On Replacement Strategies in Steady State Evolutionary Algorithms. Evol. Comput. 15, 29–59 (2007)
12. Leung, F., Lam, H., Ling, S., Tam, P.: Tunning of the Structure and Parameters of a Neural Network Using an Improved Genetic Algorithm. IEEE Trans. on Neural Networks 14, 79–88 (2003)
13. Gordon, V.S., Pirie, R., Wachter, A., Sharp, S.: Terrain-Based Genetic Algorithm (TBGA): Modeling Parameter Space as Terrain. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 229–235. Morgan Kaufmann, Orlando (1999)
14. Julstrom, B.A.: Greedy, Genetic and Greedy Genetic Algorithms for the Quadratic Knapsack Problem. In: Genetic and Evolutionary Computation Conference (GECCO), pp. 607–614. ACM, Washington (2005)